SWENG894 – Section 001
Summer 2021
Matt McIntire

CONOP – "BottomLine"

BottomLine is a web application for car shoppers looking for the best price on a new vehicle purchase. It allows customers to create an exact vehicle specification for the vehicle of their choice and obtain the best price available by submitting their configuration for car dealers to bid on. Dealers will be able to see and bid against each other for the opportunity to sell the specified vehicle at their bid price. Competition between selling dealers will result in aggressive prices for consumers.

In addition to a vehicle specification, the buyer can specify a geographical area in which they are willing to travel to complete the purchase. They can also specify optional vehicle features that they can live without to further open the bidding process to more competition from dealers.

For dealers, BottomLine represents another channel that provides immediate access to potential customers. Further, the BottomLine customer is typically toward the end of the purchase process, very likely to buy, and needs little in terms of time commitment from the dealer staff. These sales represent low-touch, quick-sale opportunities that will work particularly well for larger volume-oriented dealerships.

Problem Domain Model

Figure 1 below shows the domain model for the proposed system.

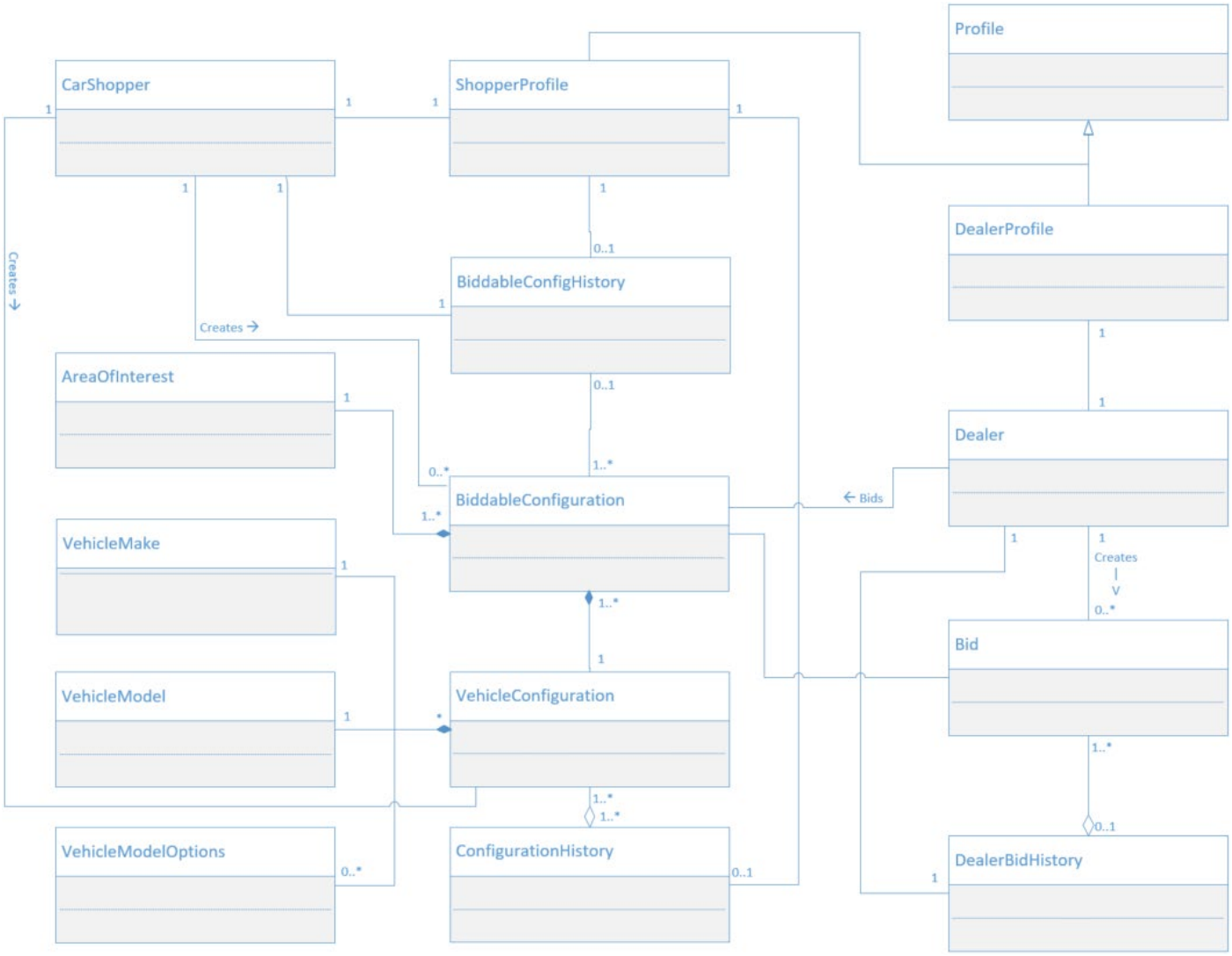

Figure 1: Domain model for the BottomLine application

Programming Language

- Python.

Django is a web application framework based on Python. Development using the framework is therefore centered around Python programming. It is very likely other languages will be required and used as part of the project (e.g. YAML, SQL), but the main programming language is expected to be Python.

Toolset

The tools listed below are considered the most likely candidates for use in the development, testing, and integration of the BottomLine project. In some cases, multiple tools are listed as possibilities. Due to the lack of expertise in any specific tool, it is anticipated that some experimentation will be needed before settling on a specific tool that best suits the preferences of the developer and the needs of the project. Therefore, this list will be refined and will narrow as the project progresses.

- MySQL-compatible database (possibly AWS Aurora or MySQL)

    - BottomLine will need storage of various types of data including vehicle, user, login, and dealer information. A relational database is a simple solution for this need as it is easily accessible programmatically via SQL, scalable, and well understood in the context of web applications.
- Code management and version control: Git (likely hosted at GitHub or GitLab)
    - Git is a widely used source code version control project and both GitHub and GitLab are widely used and provide free access for programmers to use for version control in their projects.
- Testing framework: unittest (standard in Python 3)
    - As the primary development language is anticipated to be Python, a suitable unit test framework that targets Python needs to be selected. Some brief research turned up unittest as a recurring recommendation. It is also built into Python 3 and should therefore be well-documented and not require additional integration to get set up and working.
- Deployment: Hashicorp Terraform
    - Optional. If the project gets mature enough to allow programmatic deployment to a cloud provider such as AWS, Terraform is a strong multi-cloud choice for deploying infrastructure-as-code which is not tied to any particular cloud provider.
- Web Hosting: AWS
    - As a web application, BottomLine will require some form of web hosting. AWS provides economical, scalable, and resilient hosting services and a number of different service offerings to accommodate hosting a Django-based application. Coupled with their ability to host relational databases such as MySQL and PostgreSQL, and quick setup and teardown through infrastructure-as-code provisioning tools like CloudFormation and Terraform allow for more time to be spent developing the application and less configuring, tuning, and managing the platform and underlying infrastructure.
- IDE: PyCharm, Visual Studio Code, or Jupyter Notebooks.
    - Several options have been listed here. Some brief research into modern Python editors yielded the results in the list. It is anticipated that development will start with PyCharm or Visual Studio Code.
- Web application framework: Django
    - Brief research was conducted into popular modern web application frameworks. Among the various choices, Django emerged as a viable candidate due to its foundations in Python (which meshes well with existing experience of the developer), and its focus on being high level and well suited to database-driven applications, such as BottomLine.