

Product Backlog

The table below shows the product backlog for the BottomLine app. This is a living list and will change throughout the project. Each item on the backlog has been categorized broadly and prioritized.

ID	Description	Type	Area	Priority
BL-1	As a site developer, I would like the site to make use of a modern, actively-developed, secure framework	Non-Functional	Architecture	HIGH
BL-2	As a site developer, I would like the site to use a database implementation that is modern, secure, actively-developed, and compatible with the selected web development framework	Non-Functional	Architecture	HIGH
BL-3	As a site maintainer, I need to update the vehicle make, model, and options as OEM offerings change	Non-Functional	Architecture	HIGH
BL-4	As a site user, I want to sign up for an account	Functional	Accounts	HIGH
BL-5	As a site user, I want to log into the application	Functional	Accounts	HIGH
BL-6	As a site user, I want to log out of the application	Functional	Accounts	HIGH
BL-7	As a site user, I want to be able to set and change my password	Functional	Accounts	HIGH
BL-8	As a site user, I want to be able to update my contact information	Functional	Profile	HIGH
BL-9	As a site user, I want to be able to update my contact preferences	Functional	Profile	HIGH
BL-10	As a car dealer, I want to associate myself with particular car brands I sell	Functional	Profile	HIGH
BL-11	As a car dealer, I want to specify the location(s) of my store(s)	Functional	Profile	HIGH
BL-12	As a car dealer, I need to update any of the previously saved profile information	Functional	Profile	HIGH
BL-47	As a site user, I want the ability to sign up for an account either as a car shopper or a car dealer.	Functional	Accounts	HIGH

Table 1: Product Backlog

Sprint Backlog

The table below is the backlog for the first sprint. This sprint is focused on getting the main portions of application architecture off the ground, as specified in the architectural requirements, as well as some basic application functionality that will be required for the remainder of the application to function properly. This functionality is focused on the user account request process.

ID	Description	Type	Area	Priority	UUCW
BL-1	As a site developer, I would like the site to make use of a modern, actively-developed, secure framework	Non-Functional	Architecture	HIGH	N/A
BL-2	As a site developer, I would like the site to use a database implementation that is modern, secure, actively-developed, and compatible with the selected web development framework	Non-Functional	Architecture	HIGH	N/A
BL-3	As a site maintainer, I need to update the vehicle make, model, and options as OEM offerings change	Non-Functional	Architecture	HIGH	N/A
BL-4	As a site user, I want to sign up for an account	Functional	Accounts	HIGH	15
BL-5	As a site user, I want to log into the application	Functional	Accounts	HIGH	15
BL-6	As a site user, I want to log out of the application	Functional	Accounts	HIGH	10
BL-7	As a site user, I want to be able to set and change my password	Functional	Accounts	HIGH	15
BL-47	As a site user, I want the ability to sign up for an account either as a car shopper or a car dealer.	Functional	Accounts	HIGH	N/A

Table 2: Sprint 1 Backlog

An estimate is provided on duration of the sprint. The estimation is based on Use Case Points as described by Mike Cohn¹. The UUCW (Unadjusted Use Case Weight) field in the table above is calculated using the use case definition provided in the next section of this document. Note that the UUCW only applies to the functional requirements, which have been developed into use cases in the following section. The work needed to address the non-functional architectural requirements is captured in the Technical and Environmental Complexity factors.

The total UUCW is then 55.

The first sprint will feature two human actors using the system via GUI. Their weight is therefore 3 each. In addition, the system itself interfacing with the database represents a simple actor with a weight of 1.

The Unadjusted Actor Weight (UAW) is $2 * 3 + 1 = 7$

The Unadjusted Use Case Points (UUCP) can then be calculated as $UUCW + UAW$:

$UUCP = 55 + 7 = 62$

We next calculate the project's technical complexity using the table below.

Factor	Weight	Assessment	Impact
Distributed System	2	3	6

Performance Objectives	2	3	6
End-user efficiency	1	4	4
Complex processing	1	2	2
Reusable code	1	2	2
Easy to install	0.5	1	0.5
Portable	0.5	1	0.5
Easy to change	2	4	8
Concurrent use	1	5	5
Security	1	5	5
Access for third parties	1	2	2
Training needs	1	2	2
TOTAL (TFactor)			43

Table 3: Technical Complexity Factor weighting and impact

The Technical Complexity Factor (TCF) is given by:

$$TCF = 0.6 + (0.01 * TFactor)$$

TCF is therefore:

$$TCF = 0.6 + (0.01 * 43) = 1.03$$

Next, we calculate the Environmental Complexity using the table below.

Factor	Weight	Assessment	Impact
Familiar with the development process	1.5	2	3
Application experience	0.5	2	1
Object-oriented experience	1	5	5
Lead analyst capability	0.5	3	1.5
Motivation	1	5	5
Stable requirements	2	3	6
Part-time staff	-1	2	-2
Difficult programming language	-1	1	-1
Total (EFactor)			18.5

Table 4: Environmental Factor weighting and impact

The Environmental Factor (EF) is given by the following equation:

$$EF = 1.4 + (-0.03 * EFactor)$$

EF is therefore:

$$EF = 1.4 + (-0.03 * 18.5) = 0.845$$

Note that both EF and TCF are based on global aspects of the project, while the use case points are specific to the use cases in this sprint. Therefore, as the project progresses, future sprints may see no change in the EF and TCF values used to estimate sprint duration.

To calculate the final Use Case Points (UCP), we use the following formula:

$$UCP = UUCP * TCF * EF$$

$$UCP = 62 * 1.03 * 0.845 = 54$$

Based on the UCP value and using an estimate of 20 hours per use case point, we arrive at an estimate of:

Total Hours = 54 * 20 = **1080 hours**

Functional Requirements (Use Cases)

Each functional requirement in the product backlog is described below as a use case. Acceptance test criteria are included for each requirement.

Use Case Name: Sign up for account		ID: BL-4, BL-47
Goal in Context: A new user to the system wishes to create a new account.		
Primary Actor(s): User, BottomLine system		
Preconditions: User has navigated to the BottomLine landing page.		
Scenario: <ol style="list-style-type: none">1. User selects the New User control.2. The system presents an account request form.3. The user selects the account type (dealer or shopper).4. The user enters the required information, to include email, password, name, phone number.5. The user submits the form.6. The application validates the form to determine user uniqueness (i.e. does not already have an account) and password requirements are met.7. The system creates a new user.8. The system presents a confirmation to the user on the display.		
Alternatives: 5.a The password supplied does not meet the criteria. The form is presented to the user again. 5.b The user email is already in the system. An error message is displayed to the user.		
Acceptance Criteria: A new account has been created for the user.		

Use Case Name: Log in to site	ID: BL-5
Goal in Context: A user wishes to log into the site using their account	
Primary Actor(s): User, BottomLine system	
Preconditions: The user has an active account and has navigated to the BottomLine landing page.	
Scenario: <ol style="list-style-type: none"> 1. The user selects the Log In control. 2. The system presents the login form. 3. The user enters their username (email) and password. 4. The user submits the form. 5. The system looks up the user and verifies their password. 6. The system creates a new session for the user. 7. The system indicates to the user that they are logged in. 	
Alternatives: <p>5.a The email address supplied is not found in the system. The form is presented to the user again. An error message indicating either the username OR password was incorrect.</p> <p>5.b The password supplied is not correct. The form is presented to the user again. An error message indicating either the username OR password was incorrect.</p> <p>NOTE: in either case, the same message is presented to the user.</p>	
Acceptance Criteria: The user is logged into the site and has an active session.	

Use Case Name: Log out of site	ID: BL-6
Goal in Context: A user wishes to log out of their account.	
Primary Actor(s): User, BottomLine system	
Preconditions: The user is currently logged into the system.	
Scenario: <ol style="list-style-type: none"> 1. The user selects the logout control. 2. The system destroys the session for the user. 3. The system takes the user back to the main landing page. 	
Alternatives: <p>2.a The session is already invalid. The system has nothing to destroy and follows the rest of the use case scenario starting at step 3.</p>	
Acceptance Criteria: The user is logged out of the system.	

Use Case Name: Change password	ID: BL-7
Goal in Context: A user wishes to change their account password.	
Primary Actor(s): User, BottomLine system	
Preconditions: User has an active account and knows their current password. The user has logged in and navigated to the account profile page.	
Scenario: <ol style="list-style-type: none"> 1. The user selects the change password control from the profile page. 2. The system displays a password change form. 3. The user enters their current password. 4. The user enters a new password twice. 5. The user submits the form. 6. The system validates the exiting password to confirm the current account holder. 7. The system checks the new password to see that both entries match, are different from the original, and meet the complexity requirements. 	
Alternatives: <p>6.a The current password is incorrect. The system returns the user to the password change screen and indicates the error.</p> <p>6.b The new passwords do not match, do not meet the requirements, or the hashed versions of the old and new match (indicating the password is not changing). The system returns the user to the password change screen and indicates the error.</p>	
Acceptance Criteria: The user's password has been changed.	

Use Case Name: Update contact information	ID: BL-8
Goal in Context: A user wants to change their contact information associated with their account.	
Primary Actor(s): User, BottomLine system	
Preconditions: The user is logged in to the system and has navigated to the account profile page.	
Scenario: <ol style="list-style-type: none"> 1. The user selects the edit control from the contact information section of the profile page. 2. The system makes the fields in that section editable for the user. 3. The user can make changes to the existing fields to update their settings. 4. When done, the user selects the save control to apply the updates. 5. The system validates the fields and stores the new data. 6. The system returns the user to the original profile page. 	
Alternatives: <p>5.a The email address or phone number are blank or invalid. The system returns the user to the editable contact information page and indicates the error.</p>	
Acceptance Criteria: The user has updated their contact information.	

Use Case Name: Update contact preferences	ID: BL-9
Goal in Context: A user wants to update the preference for which communication method is used to contact them.	
Primary Actor(s): User, BottomLine system	
Preconditions: The user is logged in to the system and has navigated to the account profile page. The user has more than one contact method currently stored in their user profile.	
Scenario: <ol style="list-style-type: none"> 1. The user selects the update communication preferences control from the contact information section of the profile page. 2. The system enables a list of communication methods from which the user can choose the primary method. 3. The user selects a method. 4. The user selects the save control to apply the updates. 5. The system applies the changes. 6. The system returns the user to the main profile page. 	
Alternatives: NONE	
Acceptance Criteria: The user communication preferences have been updated.	

Use Case Name: Dealer set brand affiliation	ID: BL-10
Goal in Context: A car dealer wants to set the brand(s) of vehicles for which they are affiliated.	
Primary Actor(s): Car dealer, BottomLine System	
Preconditions: The dealer has an active account on the system, is currently logged in, and has navigated to the user profile page.	
Scenario: <ol style="list-style-type: none"> 1. The dealer selects the edit control in the Dealer Information section of the profile page. 2. The system returns an editable version of the Dealer Information. 3. The dealer selects a make from a list of all vehicle makes. 4. The dealer selects Add to add it to their profile. 5. The dealer repeats steps 3 and 4 until they have added all brands they sell at their dealership. 6. The dealer selects the Save control. 7. The system updates the information and adds the make(s) to the dealer's profile. 8. The system returns the dealer to the profile page. 	
Alternatives: NONE	
Acceptance Criteria: The dealer's brand affiliation has been set	

Use Case Name: Dealer set store location	ID: BL-11
Goal in Context: A dealer wants to set the geographic location associated with the vehicle make(s) they sell.	
Primary Actor(s): Car dealer, BottomLine system	
Preconditions: The dealer has an active account on the system, is currently logged in, and has navigated to the user profile page.	
Scenario: <ol style="list-style-type: none"> 1. The dealer selects the Add Location control next to a vehicle make listed in the Dealer Information section of the profile page. 2. The system returns an address form that is editable. 3. The dealer enters the address information for the dealership for that vehicle make. 4. The dealer selects the Save control. 5. The system validates the address. 6. The system saves the address. 7. The system returns the dealer to the profile page. 	
Alternatives: <ol style="list-style-type: none"> 5.a The address is invalid. The system returns the user to the editable address form and presents an error message about the address validation failure. 	
Acceptance Criteria: The location information for the dealer's store(s) has been saved.	

Use Case Name: Dealer change profile information	ID: BL-12
Goal in Context: A dealer wants to be able to change information in their dealer profile that they entered previously.	
Primary Actor(s): Car dealer, BottomLine system	
Preconditions: The dealer has an active account on the system, is currently logged in, and has navigated to the user profile page.	
Scenario: <ol style="list-style-type: none"> 1. The dealer browses to the Dealer Information portion of the profile page. 2. The dealer selects the Edit control next to the particular item they wish to update. 3. The system returns a form which makes that item editable. 4. The dealer edits the information as needed 5. The dealer selects the Save control. 6. The system validates the information. 7. The system saves the information. 8. The system returns the dealer to the profile page. 	
Alternatives: <ol style="list-style-type: none"> 5.a The dealer decides not to save their changes. The dealer selects the Cancel control and the system returns the dealer to the Profile page. 6.a The information can not be validated. The system returns the user to the editable form and presents an error message about the validation failure. 	
Acceptance Criteria: The dealer has updated the desired fields and the system has stored those updates.	

Use Case Name: Shopper set home location	ID: BL-16
Goal in Context: A car shopper wants to set the geographic location of their home in their profile.	
Primary Actor(s): Car shopper, BottomLine system	
Preconditions: The car shopper has an account, is logged in, and has navigated to the profile page.	
Scenario: <ol style="list-style-type: none"> 1. The car shopper finds the My Locations portion of the profile page and selects the Edit control. 2. The system returns an address form that is editable. 3. The car shopper enters the address information for their desired location. 4. The car shopper selects the Save control. 5. The validates the address. 6. The system saves the validated address. 7. The system returns the car shopper to the profile page. 	
Alternatives: <ol style="list-style-type: none"> 4.a The car shopper decides not to save the address. They select the Cancel control and are returned to the profile page. 5.a The address is invalid. The system returns the car shopper to the editable address form. 	
Acceptance Criteria: The car shopper has added their home location to their profile.	

Use Case Name: Site maintainer disable account	ID: BL-14
Goal in Context: A site maintainer wants to manually disable a user account.	
Primary Actor(s): Site maintainer, BottomLine system	
Preconditions: Site maintainer has an administrative account, is logged into the admin interface, and knows the username for the user account they wish to disable.	
Scenario: <ol style="list-style-type: none"> 1. The site maintainer finds the User Lookup portion of the admin interface and enters the username for the user of interest. 2. The site maintainer selects the Search control. 3. The system performs a search for any users matching the search term. 4. The system returns a list of search results. 5. The site maintainer selects the user of interest from the list. 6. The site maintainer selects the Disable control. 7. The system disables the account. 8. The returns a message to the site maintainer indicating the account has been disabled. 	
Alternatives: <ol style="list-style-type: none"> 4.a No users are found matching the term. The system presents a message to the site maintainer indicating such. 7.a The selected account is already disabled. The system presents a message to the site maintainer indicating such. 	
Acceptance Criteria: The desired user account is disabled.	

Use Case Name: Site maintainer enable account	ID: BL-15
Goal in Context: A site maintainer wants to manually disable a user account.	
Primary Actor(s): Site maintainer, BottomLine system	
Preconditions: Site maintainer has an administrative account, is logged into the admin interface, and knows the username for the user account they wish to enable.	
Scenario: <ol style="list-style-type: none"> 1. The site maintainer finds the User Lookup portion of the admin interface and enters the username for the user of interest. 2. The site maintainer selects the Search control. 3. The system performs a search for any users matching the search term. 4. The system returns a list of search results. 5. The site maintainer selects the user of interest from the list. 6. The site maintainer selects the Enable control. 7. The system enables the account. 8. The returns a message to the site maintainer indicating the account has been enabled. 	
Alternatives: <ol style="list-style-type: none"> 4.a No users are found matching the term. The system presents a message to the site maintainer indicating such. 7.a The selected account is already enabled. The system presents a message to the site maintainer indicating such. 	
Acceptance Criteria: The desired user account is disabled.	

Architecture

An initial architecture diagram for the BottomLine application is shown below in Figure 1. BottomLine will be web-based and implemented using the Django framework. The backend will feature a database for storage and retrieval of user credentials, profile data, vehicle information, and bid details. If possible, a third-party solution will be utilized for retrieval of vehicle data (e.g. make, model, options) to reduce the burden of maintaining and updating that information locally in the BottomLine backend.

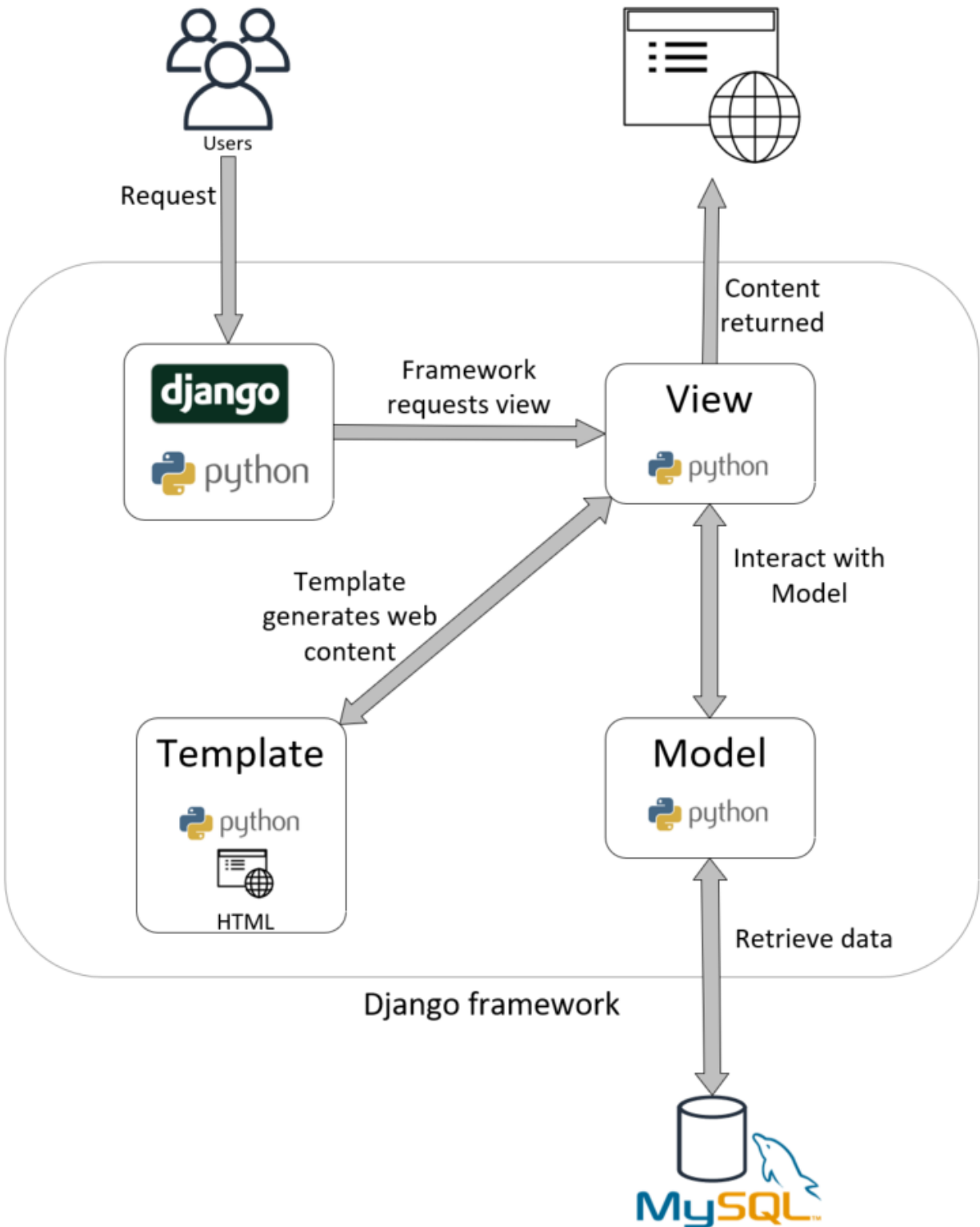


Figure 1: BottomLine deployment diagram

The Django framework is designed to implement a self-described Model-View-Template (MVT) architecture and provides robust support for several database implementations which it abstracts

through a data model based on Python mappings between data types. The difference between MVC and MVT in the case of Django is that Django largely implements the Controller portion itself (i.e. it is handled transparently for the developer) and instead relies on template files to render the HTML to the user. In their opinion, the View is not strictly the rendered content on the webpage presented to the user, but the response from the server based on the user's request. The rendered web content is derived from the response being passed into one or more templates.

Django's strengths are that it is being actively developed, which means bugs and security issues are being resolved in a timely fashion. It also has strong security provisions, providing protections for XSS, CSRF, SQL injection, and clickjacking out of the box. Django is also Python-based, which is a good fit for the BottomLine developer skillset at present. Finally, it is a popular framework and has an active user community, many tutorials, and excellent documentation. These features all help the developer in the course of deploying a robust, secure web application.

Django also supports most modern relational database technologies. BottomLine will require a relational database in some form for storage and manipulation of a variety of data related to the vehicle configuration, bidding, and other business logic within the application.

Within the scope of the application, a context diagram is shown in Figure 2 below. Car shoppers and car dealers interact to facilitate the sale of a vehicle using BottomLine to accomplish this primary goal. Site Maintainers are also shown as they have requirements that need to be addressed in the context of the application in order to ensure its security and availability.

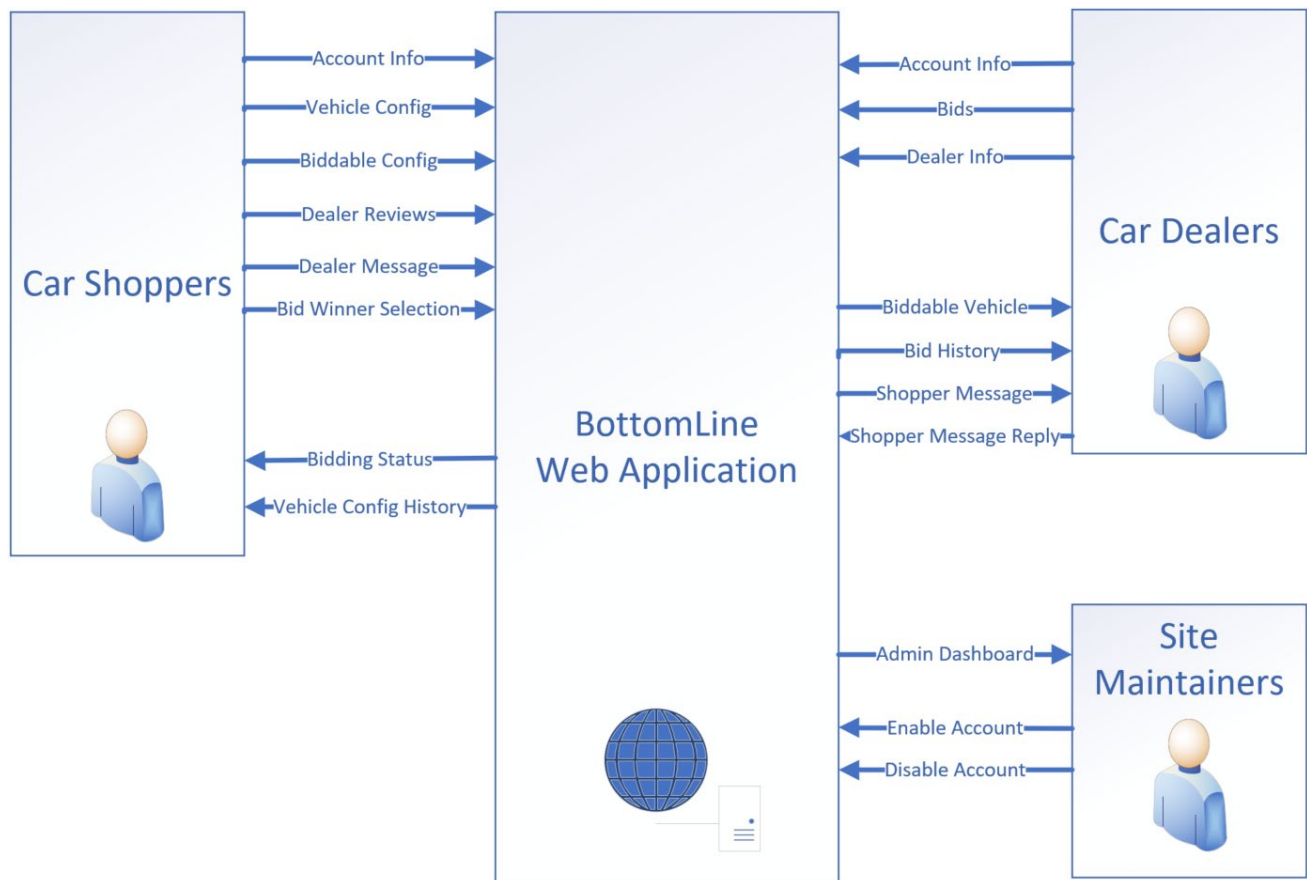


Figure 2: BottomLine context diagram

References

¹Cohn, Mike. "Estimating with Use Case Points". Mountain Goat Software. Retrieved 23 May 2021 from: [https://courses.worldcampus.psu.edu/canvas/su21/2215min-4381/content/02_lesson/corefiles/Cohn - Estimating with Use Case Points v2.pdf](https://courses.worldcampus.psu.edu/canvas/su21/2215min-4381/content/02_lesson/corefiles/Cohn_-_Estimating_with_Use_Case_Points_v2.pdf)