AuctioX – Documentation

Baisan Razvan, Ciulin Adrian, Hirtobanu Gabriel, Vatamanu Vlad

Group A3

Architecture – Modules

RequestHandler  
 RequestHandler is a subtype of BaseHTTPRequestHandler that, as the name suggests, handles HTTP requests and forwards request information to the Dispatcher.

Dispatcher  
 The Dispatcher will provide most of the server's logic. This includes authentication, creating users, buying, selling, posting questions, reviews, banning users, matching views etc.

User  
 The User class is a simple object that encapsulates the data of any registered user and provides user manipulation functionality such as buying, selling, reviewing or confirming payment.

Admin  
 The Admin is a subtype of a User. This class has additional rights and methods for general administration and moderation of the website.This includes banning, sending notices to users etc.

Report  
 The Report class is an object meant for manipulating the corresponding database table.

Notice  
 The Notice class is an object meant for manipulating the corresponding database table.

Product  
 The Product class models an item listed for auction, selling or an archived item (that was removed from public listing).

This class models the condition, price, date of listing and other such characteristics of the product.

View  
 The View class is an interface to all the functions that can generate a response coresponding to the received request.

StateExporter  
 The StateExporter class handles any type of request for a news stream.

Technology stack

# Python [Programming language]

The Python programming language is well known for its ease of use and because it’s great for rapid prototyping and iteration. Two of the most important features that we took into consideration when choosing were its minimal setup procedure and the code’s readability.

# SocketServer [Python module]

The SocketServer module simplifies the task of writing network servers. The SocketServer.TCPServer class uses the Internet TCP protocol, which provides for continuous streams of data between the client and server.

# BaseHTTPServer [Python module]

This module defines the classes for implementing HTTP servers and it is the basis for building the Web server. We use the BaseHTTPServer.BaseHTTPRequestHandler class to handle the HTTP requests that arrive at the server. The handler will parse the request and the headers, then call a method specific to the request type.

# urlparse [Python module]

This module defines a standard interface to break Uniform Resource Locator (URL) strings up in components. We are using this module in order to ease parameter extraction from the URL.

# mimetypes [Python module]

The mimetypes module converts between a filename or URL and the MIME type associated with the filename extension. The module provides a convenient way to match a requested file with its correct MIME type.

# os, shutil [Python module]

The os and shutil modules provide us with a basic file I/O system. They facilitate interaction with individual files from the disk.

# re [Python module]

This module provides regular expression matching operations. We use this module when dispatching a request to it’s appropriate view.

# Jinja2 [Python module]

Jinja2 is a full featured template engine for Python. Essentially, templates can contain variables as well as some programming logic, which when evaluated (or rendered into HTML) are replaced with actual values. We plan on using this module for page generation since we believe that page generation by hand or directly through code is highly inefficient and a nightmare to perform maintenance on.

# pyPDF2 [Python module]

This is a module which provides us a way to write PDF files and we will use it to export data in this format.

# json [Python module]

This Python module will allow us to export data as JSON format.

# xml [Python module]

This package contains interfaces for processing XML formatted data. We will use it in order to export data in this format.

# hashlib [Python module]

This package contains hashing functions. We plan to use this module for encrypting sensitive user data.

# MariaDB [Database engine]

This is a database engine that directly competes with MYSQL and as of now, it appears to surpass it.

We have chosen to use it for a variety of reasons:

* More cutting edge features such as dynamic column support which allows us to get access to both SQL and NoSQL in the same database.
* A bigger variety of storage engines such as CONNECT which allows access to different kinds of text files and remote resources as if they were regular MariaDB tables and others and they come with the official release thus, there is no need to use 3rd parties
* MariaDB claims it has a much improved query optimizer and many other performance related improvements. Certain benchmarks show that MariaDB is radically faster than MySQL. Benchmarks don’t however always directly translate to real life situations. Faster is always better, even if it is just a bit faster.
* Less bugs and compare to other DB engines, easier debugging in case of something wrong.

# mxODBC [Python module]

This package is a Python API for ODBC compatible databases. We will be using this one because according to the wiki, it supports Python 2.7 while other APIs don't. It is also proprietary software and likely to be more stable.

User actions

# Everyone (these actions can be performed by all of the following user types)

Search – The user can choose to do a simple or an advanced search. After inputting a string, they are presented with a list of items relevant for their search. Advanced search adds the option to filter the results. The filters can also be changed on the results page.

User profiles – Accessing the profile pages of users who’ve created a listing. These will display data such as the number of transactions made and the history and overview of the feedback they’ve received from other users.

Report – Creates a ticket to report a certain user or a listing; a reason will be selected from a list and more details will optionally be given.

# Users who are not logged in

Register – Creates account.

Forgot password – Request an email containing login information for an email address associated with an existing account; provides a link to a page user for changing the password.

Sign in – Logs into an account.

# Users who are logged in

Buy – (for Buy-it-now type items) The user places an order for an item. After inputting and confirming the payment data, the transaction is finalized.

Bid – (for Auction type items) The user places a bid (higher than the current one) for an item. The user can place another bid later on in case they’ve been outbid. After the item listing time expires, the user pays for the product.

Sell – Creates a new listing. The user must provide a picture and details about the item they want to sell and select the listing type. For the Buy-it-now type they will add a price, and the Auction type requires a starting price and the listing duration.

Account settings – The page facilitates the deletion of listings created by the user, changing the password and viewing the transaction and feedback history.

Ask question – The user can add a public question for a certain listing.

Add feedback – User adds feedback for their transaction with the seller.

Ask for refund – User asks for a refund for a product they paid for but didn’t receive. After a certain amount of time has passed, the user can report the seller for fraud.

Log out – Signs out of an account.

# Administrator

Ban account – Bans a user account that broke the site rules. The reason for deletion will be specified and a notification email will be sent to the user.

Remove listing – Removes a listing that breaks the site rules. The reason for deletion will be specified and a notification email will be sent to the user. After a certain number of deleted listings, the user will be banned.

Report overview – The administrator is presented a list of report tickets submitted by users. The administrator can validate or invalidate them and take the appropriate course of action. Too many invalid tickets sent by a user will lead to a notification email, and eventually a ban for the user who sent them.

Solved tickets overview - The administrator is presented a list of report tickets that have been marked as solved, ordered by the date at which they’ve been solved.

Log out – Signs out of the account.

Tasks

Database & Integration – Hirtobanu Gabriel

User Management – Ciulin Adrian-Mihai

Product Management, Server – Razvan Baisan

Administration system – Vatamanu Vlad