

CMU 17637 - Web Application Project Specification

Easybid: A goods auction website

Xiangyu Wu (xwu4), Congtao Kuang (congtaok), Haobin Tan (haobint)

Description

We are building a website for shopping in an auction or bidding model. On the website, seller users can publish the goods they want to sell, determine the time period of the auction, the starting price, the direct purchase price, and the minimum markup price, etc. Buyer users can browse the goods being auctioned, join the auction and give a willing price. If the highest price is maintained during the auction period of the product, the product will be purchased at the highest price. If the price of the product is exceeded by other buyer users' bids during the auction period, the exceeded user will be notified and invited to bid again.

Product backlog: Functionalities

- Pages / templates
 - Home page
 - Can be viewed without login
 - All auction can be viewed
 - Sorting function might be available (option)
 - Basic information for each auction will be displayed:
 - Product image
 - Current Price
 - Close Time
 - Login/ register option available on top right
 - Login/register page
 - Django user authentication login/register
 - OAuth to support different way for login/register (optional)
 - Product viewing page (for each auction product)
 - Product image display
 - Bidding information display:
 - Current Bid price (will be real time dated by website)
 - Auction End countdown
 - Number of bids
 - Current highest bid username
 - Placing new bid button

- Only available for logged user
 - Non-logged user will see button for register/login
 - Product description
- Selling page
 - Create auction form:
 - Auction title/name
 - Description of the auction product
 - Auction product picture upload (if none then will display default image)
 - Minimum value
 - Step value (minimum increasing amount for each new bid)
- Biding logistic - WebSocket based communication between clients and server
 - Communication protocol to prevent bidding confliction and achieve fault-tolerance
 - Clients to server message packet:
 - client ID
 - sequence number
 - biding price
 - authorization credential
 - e.g.: "1 2 67 F4FCFF"
 - Server to clients message packet, should send to all clients that attending the biding:
 - current highest price
 - current highest price user ID
 - biding notification: bid successfully/failed (optional, when received client bid request)
 - outbid notification: (optional, when client's biding price is no longer the highest one)
 - e.g.: "67"

Sprint #1 Backlog

- Registration/Login page with Django authentication - Haobin
- Home page - Haobin
- Product view (show websocket working for bid update) - Congtao
- Model, forms - Sherry
- Setup- Sherry

Product owner

Congtao Kuang for Srint #1

Data Model

- User (django default authentication)
 - username
 - first_name
 - last_name
 - email
 - password
- Profile
 - owner (OneToOneField)
 - bio_text(CharField)
 - profile_picture (FileField)
 - content_type (Charfield)
- AuctionItem
 - id (primary key)
 - property: is_expired (BooleanField)
 - product_name (CharField)
 - seller (Foreign key:user)
 - product_value (FloatField, min = 0)
 - starting_price (FloatField, min = 0)
 - min_bid_increment (FloatField, min_val = 1)
 - final_price (FloatField)
 - start_time (DateTimeField)
 - end_time (DateTimeField)
 - product_image (Filefield)
 - content_type (CharField)
 - product_description (CharField)
 - bid_count(int field)
 - property: highest_bid (Foreign key:Bid)
 - property: time_remaining (DateTimeField)
- Bid
 - id (primary key)
 - auction_item (Foreign key: AuctionItem)

- bidder (Foreign key: user)
- time (DateTimeField)
- price (IntegerField)

Possible Technologies

- Bootstrap CSS framework - for a more colorful and modern user interface
- AWS EC2 - for deploy purpose
- MySQL - for data storage
- Django framework - the main framework that all members are familiar with
- React framework - will be used with django to build a more user interactive website
- Websocket - updating current bid price for buyers and sellers
- OAuth - different ways for users to register
- Email with Django - sending notifications to buyers and sellers about current status of the goods
- Payment - for buyers to pay and sellers to receive payment (optional)

UI Mockup

Referenced in figma project

<https://www.figma.com/file/qF9AHZYetTJViiSvs5Bb4C/EasyBid?node-id=0%3A1>