# Bezos Auctions

# EECS 4413 - Project Deliverable 2

| | |
|---|---|
| Version: | 1.0.0 |
| Print Date: | 2023-11-13 |
| Release Date: | 2023-11-13 |
| Release State: | Final |
| Approval State: | Approved |
| Approved by: | Kakshil Patel, Arjun Kaura, Yash Rathore, Kavian Nasseri |
| Prepared by: | Kakshil Patel, Arjun Kaura, Yash Rathore, Kavian Nasseri |
| Reviewed by: | Kakshil Patel, Arjun Kaura, Yash Rathore, Kavian Nasseri |
| Path Name: | …/EECS4413_Asg2/EECS4413_Asg2_document.pdf |
| File Name: | EECS4413_Asg2_document.pdf |
| Document No: | 1 |

## Document Change Control

| Version | Date | Author(s) | Summary of Changes |
|---|---|---|---|
| Deliverable 2 | Nov 1, 2023 | Kakshil, Arjun, Yash, Kavian | Got in a group call, discussed the plans, assigned tasks to each group member and started the work |
| Deliverable 2 | Nov 4, 2023 | Kakshil, Arjun, Yash, Kavian | Created the design and overall look of the different pages |
| Deliverable 2 | Nov 7, 2023 | Kakshil | Implemented the frontend and backend for the welcome, sign up/login mechanism, sessions |
| Deliverable 2 | Nov 9, 2023 | Arjun | Implemented the search and main pages frontend and backend, which included searching for an item in db and adding an item into the auction |

| | | | |
|---|---|---|---|
| Deliverable 2 | Nov 10, 2023 | Kakshil, Arjun, Kavian | Worked on implementing the forward and dutch auctions, selecting an item from the auction items table, which takes you to the item details page, in which you can bid |
| Deliverable 1 | Nov 10, 2023 | Yash | Implemented auction ended and payment front end and backend mechanism, specific to each user. |
| Deliverable 1 | Nov 11-12, 2023 | Arjun, Kakshil, Yash | Implemented Bidding page to enter bid amount |
| Deliverable 1 | Nov 13, 2023 | Arjun, Kakshil, Yash, Kavian | Finalised report – proofreading, adding diagram. Checked over code and fixed any bugs. |

# Document Sign-Off

| **Name (Position)** | **Signature** | **Date** |
|---|---|---|
| Kakshil | KP | 2023-11-13 |
| Arjun | AK | 2023-11-13 |
| Yash | YR | 2023-11-13 |
| Kavian | K | 2023-11-13 |

# Installation Instructions

1. Download the zip file containing the project, databases, curl/postman commands and design document.
2. Import the project into eclipse as WAR, it is titles “Bezos.war”
3. Use the register.db file in the root directory of the submission folder
4. Go to the context.xml and write down the address to which the database is located eg. "jdbc:sqlite:/Users/kakshilpatel/Desktop/register.db"
5. Click on project and “run on server”, it should take you to the welcome page, and you can test from there.

## Implementation choices and their justification

We chose to use a SQLite database that stores 3 tables.

- The first table is the ‘users’ table with all the fields containing the user information, username and password. This is accessed using sign in/login, session implementation as well payment details.
- The second table is the ‘items’ table with the fields containing current item value, auction type, time remaining, description and shopping. This information is accessed in the search page, and well as bidding and payment implementation.
- The last table is the ‘bids’ table with the fields containing the user information and the bid information to keep track off which user bought what item.

We implemented an option to add an item to the auction pool buy using a page. This is to make it easier for the user to add new items, and easier for us to test. This also allows for users to sell a bought item again by adding it back into the auction UC7.

## Design Choices

We broke down each page's Servlet into its own servlet class. All the features of a particular page are within that single servlet class associated with it. This is following the singleton pattern, which we outlined in our first report.

There is also the use of code modularity, as each servlet on its own does not contain the feature or references to other servlets. They are independent and linked only to their associated html page.

Below is a diagram of the Use Case 2, which is the User Login. We followed a simple approach of verifying information through the database and limiting the number of session variables.

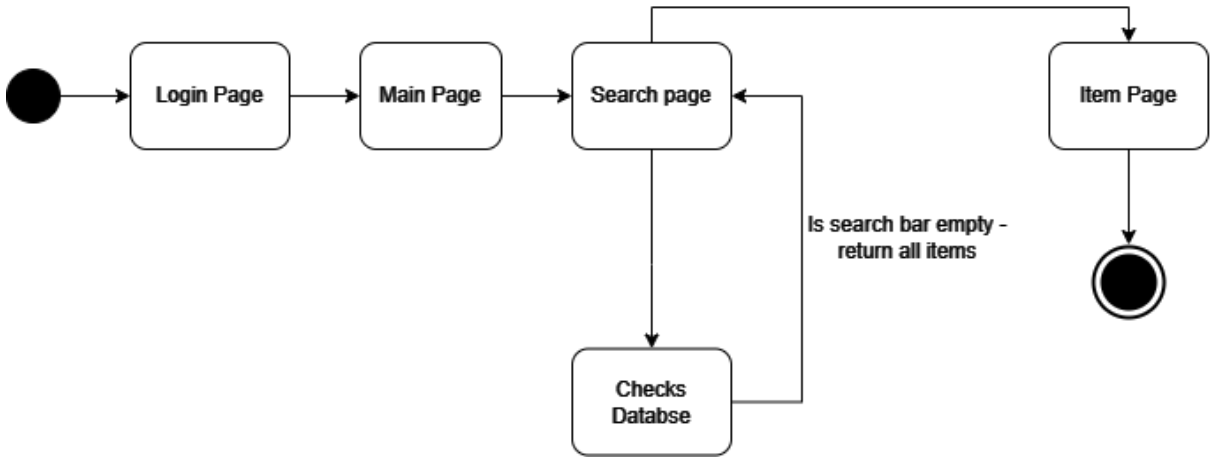


**Figure 1: Activity Diagram of Use Case 2**

We followed this original plan very closely for other systems too such as the sign up feature.

# Test Use Cases

**UC1:**

| | |
|---|---|
| Test ID | 0 |
| Category | Evaluation of user credentials stored on DB |
| Requirements Coverage | UC1-Successful-User-Login |
| Initial Condition | Database operational |
| Procedure | 1. User selects login<br>2. User provides username<br>3. User provides password<br>4. User logs in and next page is shown |
| Expected Outcome | Login page closes, user is presented with the main page of options |
| Notes | the user provides only alphanumeric usernames, and passwords without any spaces |

| Test ID | 1 |
|---|---|
| Category | Evaluation of user credentials stored on DB |
| Requirements Coverage | UC1-Unsuccessful-User-Login |
| Initial Condition | Database operational, accounts already present in DB |
| Procedure | 1. User selects login<br>2. User provides username not present in database<br>3. User provides password<br>4. User logs in and warning message is shown |
| Expected Outcome | Login page is shown again with a message |
| Notes | the user provides a username that is not present in the database, and/or a password with spaces. Since the account with the username does not exist in the database, it throws an error exception |

| Test ID | 2 |
|---|---|
| Category | Evaluation of user credentials stored on DB |
| Requirements Coverage | UC1-Successful-Signup |
| Initial Condition | Database operational |
| Procedure | 1. User selects signup<br>2. User provides username not present in database<br>3. User fills out email, location<br>4. User provides a password<br>5. User signs up and is sent to main page |
| Expected Outcome | User account is registered and main page is shown |
| Notes | the user should provide a username that is not present in the database and which has no spaces in it. All other details (address, email) are valid: address cannot be blank, email must have no spaces. |

| Test ID | 3 |
|---|---|
| Category | Evaluation of user credentials stored on DB |
| Requirements Coverage | UC1-UnSuccessful-Signup |
| Initial Condition | Database operational, accounts present in database already |

| | |
|---|---|
| Procedure | 1. User selects signup<br>2. User provides username present in database<br>3. User fills out email, location<br>4. User provides a password<br>5. User signs up and is sent to back to signup page with warning |
| Expected Outcome | User account is not registered and signup page is shown again |
| Notes | the user provides a username that is present in the database. All other details (address, email) are valid: address cannot be blank, email must have no spaces. |

**UC2:**

| | |
|---|---|
| Test ID | 4 |
| Category | Searching through the database for Auctioned items |
| Requirements Coverage | UC2-Successful-item-Search |
| Initial Condition | Database operational, item present in database already |
| Procedure | 1. User enters item name in the input bar<br>2. User clicks search<br>3. The auctioned item is displayed with the specified details |
| Expected Outcome | The correct auctioned item is displayed with the details specific to what the user searches for. |
| Notes | |

| | |
|---|---|
| Test ID | 5 |
| Category | Searching through the database for Auctioned items |
| Requirements Coverage | UC2-Successful-item-Search-Blank |
| Initial Condition | Database operational, item(s) present in database already |
| Procedure | 1. User leaves search bar blank<br>2. User clicks search<br>3. All bid item are listed |
| Expected | All items in the database are shown. |

| | |
|---|---|
| Outcome | |
| Notes | User must leave search bar blank |

| | |
|---|---|
| Test ID | 6 |
| Category | Selecting a bid for the searched item |
| Requirements Coverage | UC2-Successful-Item-Search |
| Initial Condition | Database operational, items shown on search page after successful search |
| Procedure | 1. User selects the radio button next to item and clicks bid<br>2. The auctioned item is displayed with the specified details |
| Expected Outcome | The correct auctioned item is displayed with the details specific to what the user searches for. |
| Notes | If the user decides to select another item, while already having an item selected, the old item will get deselected and the new item clicked will be selected as only 1 item can be selected at one time. |

| | |
|---|---|
| Test ID | 7 |
| Category | Selecting a bid for the searched item |
| Requirements Coverage | UC2-Successful-Item-Search |
| Initial Condition | Database operational, items shown on search page |
| Procedure | 1. User does not select any item to bid, all radio button blank<br>2. User selects bid button |
| Expected Outcome | User is redirected to same page, search page, and message is shown saying to select an item before bidding |
| Notes | N/A |

**UC3:**

| | |
|---|---|
| Test ID | 8 |

| | |
|---|---|
| Category | Selecting a forward bid item from search results, bidding higher amount than current bid |
| Requirements Coverage | UC3-Successful-Item-Bid-Forward |
| Initial Condition | Database operational, items shown on search page, one item must be forward auction item that is still active |
| Procedure | 1. User selects a Forward auction item<br>2. Presses Bid button, is taken to item page<br>3. User enters bid greater than amount currently bid<br>4. User selects bid button |
| Expected Outcome | User is shown the message “Bid Successful” and the new bid amount is shown on the item page. In the backend, the bid amount is updated to the new one the user entered. |
| Notes | If current bid is at $100, user must enter more than that, such as $120 |

| | |
|---|---|
| Test ID | 9 |
| Category | Selecting a forward bid item from search results, bidding lower/same amount than current bid |
| Requirements Coverage | UC3-UnSuccessful-Item-Bid-Forward |
| Initial Condition | Database operational, items shown on search page, one item must be forward auction item that is still active |
| Procedure | 1. User selects a Forward auction item<br>2. Presses Bid button, is taken to item page<br>3. User enters bid less than amount currently bid<br>4. User selects bid button |
| Expected Outcome | User is shown the message “Bid Unsuccessful, enter greater amount” and the bid amount is not updated. |
| Notes | If current bid is at $100, user must enter less than that, or the same amount |

| | |
|---|---|
| Test ID | 10 |
| Category | Selecting a dutch bid item from search results, buying the item |

| | |
|---|---|
| Requirements Coverage | UC3-Successful-Item-Bid-Dutch |
| Initial Condition | Database operational, items shown on search page, one item must be dutch auction item that is still active |
| Procedure | 1. User selects a Dutch auction item<br>2. Presses Bid button, is taken to item page<br>3. User presses buy |
| Expected Outcome | The page is refreshed. User is shown the message “Buy successful”. and the option to Pay Now is shown on the item page. |
| Notes | |

| | |
|---|---|
| Test ID | 11 |
| Category | Auction ended for forward bid item |
| Requirements Coverage | UC3-Auction-End-Forward |
| Initial Condition | Database operational, items shown on search page, one item must be nearing its end |
| Procedure | 1. User is on page where countdown is almost done for Forward bid item<br>2. The bid ends, user clicks on item<br>3. A new page will be shown |
| Expected Outcome | The item page is shown but no details are given except “Auction has ended for item.” The Pay Now option is not there because the user has not bid on it. |
| Notes | The Pay Now option is not there because the user has not bid on it. Details for the item are not shown either because user has not won the item |

**UC4:**

| | |
|---|---|
| Test ID | 12 |
| Category | Paying for bid item |
| Requirements Coverage | UC4-Successful-Pay-for-Item |
| Initial Condition | Database operational, user won the bid or successfully bought the item if dutch auction |
| Procedure | 1. User selects the bid item<br>2. User clicks on “Pay Now”<br>3. User redirected to Payment page |
| Expected Outcome | User is redirected to Payment page |
| Notes | N/A |

| | |
|---|---|
| Test ID | 13 |
| Category | Paying for a bid item for which the user has not won. |
| Requirements Coverage | UC4-Unsuccessful-Pay-for-Item |
| Initial Condition | Database operational, user did not win the bid |
| Procedure | 1. User does not bid for an item<br>2. User clicks on “Pay now”<br>3. Failure Notice is displayed as the user has not won the bid so he cannot bid |
| Expected Outcome | Failure Notice is displayed |
| Notes | N/A |

**UC5**

| | |
|---|---|
| Test ID | 14 |
| Category | Submit payment details |
| Requirements Coverage | UC5-Successful-Payment |
| Initial Condition | Database operational, user bought item, database fills information such as Name, Last Name, street, and cost |
| Procedure | 1. User fills all payment details: Card number, Name on card, Expiry date, security code<br>2. User presses |
| Expected Outcome | Receipt page is shown. |
| Notes | N/A |

| | |
|---|---|
| Test ID | 15 |
| Category | Selecting a bid for the searched item |
| Requirements Coverage | UC2-Successful-Item-Search |
| Initial Condition | Database operational, items shown on search page after successful |
| Procedure | 1. User does not select any item to bid, all radio button blank<br>2. User selects bid button |
| Expected Outcome | User is redirected to same page, search page, and message is shown saying to select an item before bidding |
| Notes | N/A |

**UC6**

| | |
|---|---|
| Test ID | 16 |
| Category | Winning bidder Receipt and shipment info |

| | |
|---|---|
| Requirements Coverage | UC6-Successful-payment of winning bid |
| Initial Condition | Database operational, item payment is done, |
| Procedure | 1. Receipt page is shown |
| Expected Outcome | Receipt page shown. The details shown must be accurate: amount paid, amount of days, user details. |
| Notes | N/A |