**Online Auction Services**

**Advanced Software Engineering Project**

**Dr. Neeraj Gupta**

Developer: Mahmoud Zamani

Executive Master of Software Engineering

Fall 2018

**Table of Contents**

1 Introduction 3

1.1 Introduction 3

1.2 Architecture 3

2 Technologies 4

2.1 Technologies Considered: 4

2.2 Technologies Used 4

2.3 List of packages used 5

3 Functionalities 6

3.1 Non Functional Requirements 6

3.2 Functionalities 6

4 Website 8

4.1 Frontend Website: 8

5 SUMMARY 9

5.1 Problems Encountered 9

5.2 Conclusion 9

# Introduction

## Introduction

A responsive and scalable real-time online auction web application for selling/purchasing items was developed as part of our Web Programming Languages Coursework through which user can place a bid for an available item and can place an item for auctioning. It adheres to the “Service Oriented Architecture” specification. The detailed architectural diagram showing the interaction between various components is shown below

## Architecture

A close up of a map

Description automatically generated

# Technologies

## Technologies Considered:

* **Programming technologies:** Python, .net, JavaScript

As all team members were comfortable with JavaScript, so JavaScript was used for programming.

* **Runtime Environment:** NodeJS, SpiderMonkey

NodeJS was used as it is the most powerful JavaScript runtime environment and very good community support

* **Database**: MySQL, MongoDB

As for our application we don’t really need any relational DB and NoSQL DB better emulate our requirements, so MongoDB was used

* **Cache**: Redis, Memcached
* **Queues**: RabbitMQ, ActiveMQ, Redis Simple MQ  
  Implementing Redis Simple MQ was easier than RabbitMQ and ActiveMQ.
* **Web Services:** SOAP, Restful

As we are not using our APIs for other interfaces and Rest APIs are easy to implement, we are using Restful APIs for our web application

## Technologies Used

|  |  |
| --- | --- |
| **Back End** | **Front End** |
| Node |  |
| Express |  |
| ActiveMq |  |
| MemCached |  |
| MongoDb |  |
| RESTful |  |

Table 1: Technology Used

## List of packages used

|  |  |
| --- | --- |
| **Front End** | **Back End** |
|  |  |
|  |  |
|  |  |
|  |  |

# Functionalities

## Non Functional Requirements

**Client-Server Communication Encryption:** TLS/SSL has been used to encrypt communication between the client (i.e. browser), web site server, Web Services, and Microservices.

**Compression:** Requests between all the web servers are compressed and send forward using gzip

**Authentication:** Authorization key is added in the header parameter for the communication between servers.

**MemCached:** Caching technology is used in front end for quick response.

**ActiveMQ**: Queuing is done for post request (mainly bidding) so that resource usage is optimized.

Accessible any unavailable page should retrieve a pretty and generic 404 page

## Functionalities

**New user registration:** A new user is able to register by filling out the registration form to create an account. Upon successful registration the user will be redirected to the login page.

**Existing user login and logout:** Login and Logout services is provided to user, so that after logging in user can access his/her account. And during any time, user can logout.

**User profile information display and editing:** The user can view /edit their profile information.

**Forgot Password:** Password reset functionality is provided to the user.

**Ability to post an item:** A user can post an item by providing following detail:

1. Item title
2. Item description
3. Initial/minimum bidding price
4. Selecting an available auction date and time
5. Uploading item photo

**Delete a posted item:** A user can delete their posted item.

**Bid for an item**: A user can bid for any item posted by other users.

**Ability to view auction schedule:** A user can view one month of auction schedule and different items available for bidding for a particular date.

**Ability to search for items that are available for bidding during a date range:** User can select a date range to see all the item posted for auction during those dates

**Ability to filter:** A user can filter the item list and sort them on different fields

**Ability to view the summary:** A user can view the summary of all the bids for an item posted by that user.

**Admin functionalities**

**Admin Login and Logout:** A login and logout functionality is provided for Admin

**Specify the auction schedule/hours:** An admin user can specify the schedule auction hours for the day

# Website

## Frontend Website:

# SUMMARY

## Problems Encountered

## Conclusion

During the course of the project I have learnt the concept of “Service Oriented Architecture” and “Microservices Development and Architectures” and how to build a large-scale web application which is responsive and scalable, horizontally and vertically. The technologies which I have learnt during the course were very useful for implementation for the project and will be helpful for our further work in this area.