# Clients

## Android Phone Specifications

|  |  |
| --- | --- |
| Specification | Estimated Targeted Value |
| Operating System | Android 4.4 (KitKat) |
| RAM | 4GB RAM |
| CPU | 8 Core @ 1.8Ghz per core |
| Storage | At least 2 GB internal storage for cache for images and rendered model data  1GB reserved cache for 3D Models  Target is 100 cached photos  100 \* 55.84 Mb per photo = 5,584 Mb ~ 698 MB |
| Cellular Connectivity | 4G Cellular up to 15 Mbps |
| Wireless | 2.4 GHZ @ Wireless N with min 150 Mbps |
| Internet Connectivity | 30 Mbps |
| Camera | Single camera lens @ 12 Megapixel |
| Photo Size | 4290x2800 (~12MP)  PNG Compression Approx. 6.98 MB per photo or  55.84 Mb per photo |
| Photo Transmission Time | < 3 Second @ 30 Mbps Internet Speed  < 6 Seconds @ 15 Mbps Cellular |

## Computer Specifications

|  |  |
| --- | --- |
| Specification | Estimated Targeted Value |
| Operating System | Linux, Windows, Mac OS  Any operating system that supports current browsers |
| Browser | Edge, Chrome, Firefox, Safari  Browsers must support ES6 or above and HTML 5  Firefox 78 and Above  Chrome 84 and Above  Edge 83 and Above |
| RAM | 4GB RAM |
| CPU | 2 cores @ 2GHZ |
| Storage | 1 GB Cache for images and website content |
| Wireless | 2.4 GHZ @ Wireless N with min 150 Mbps |
| Internet Connectivity | 30 Mbps |

# Servers and Cloud

## Blob Storage

|  |  |
| --- | --- |
| Specification | Estimated Targeted Value |
| Platform | Amazon EFS (Elastic File System) |
| Storage Requirements | Assuming 500,000 persisted moves at 200 photos for inventory  1 photo = 55.84 Mb  200 photos = 11,168 Mb  @ 500,000 Customers  6.226 \* 10^5 TB |
| Iops | Capable of max 100,000 \* 55.84 Mb in under 5 seconds  (be able to store a photo for 100,000 simultaneous moves in under 5 seconds)  Approx. 139.6 GB/s |

## SQL

|  |  |
| --- | --- |
| Specification | Estimated Targeted Value |
| Platform | Amazon RDS (Relational Database System)  Or Aurora? |
| Vendor | MySQL |
| Memory | 128GB with scale up capabilities |
| Storage Requirements | Assuming 20 MB per customer of raw SQL data  Approx. 10 TB database Minimum size @ 500,000 customers |
| CPU | 128 cores @ 4GHZ |
| Iops | Assuming 100,000 simultaneous users  100,000 \* 0.5 MB retrieved data/sec = 50GB read/writes/sec |

## Web Services (Web API for Android Client and Website)

|  |  |
| --- | --- |
| Specification | Minimum Targeted Value |
| Platform | Amazon Elastic Beanstalk |
| Memory | 64GB with scale up capabilities |
| Storage Requirements | 100 MB < All Target Compiled Files < 500 MB |
| CPU | 32 cores @ 4GHZ with auto scale up capabilities |
| Iops | Assuming 100,000 simultaneous users  100,000 \* 0.5 MB retrieved data/sec = 50GB capability of data retrieval (depends on the database Iops as well) |

## Web Application (For Clients Accessing Through a Browser)

|  |  |
| --- | --- |
| Specification | Minimum Targeted Value |
| Platform | Amazon Elastic Beanstalk |
| Memory | 32GB with scale up capabilities |
| Storage Requirements | 100 MB < All Target Compiled Files < 500 MB |
| CPU | 16 cores @ 4GHZ with auto scale up capabilities |
| Iops | Assuming 20,000 simultaneous users  20,000 \* 0.1 MB retrieved data/sec = 2GB /sec capability of data retrieval |

## Vendor AP Bridge

|  |  |
| --- | --- |
| Specification | Minimum Targeted Value |
| Platform | Amazon Lambda |
| Memory | 16GB with scale up capabilities |
| Storage Requirements | 10 MB < All Target Compiled Files < 50 MB |
| CPU | 8 cores @ 4GHZ with auto scale up capabilities |
| Iops | Assuming 1 GB of data from each rental vendor @ 4 vendors  4GB data in < 30 seconds = 133 MB/s |

## Solution Deliverables

Android App

* Distributed via android store
* Communication is HTTP REST API with backend

Website App

* Published online for browser interaction
* Communication is HTTP REST API with backend

References

<https://design215.com/toolbox/megapixels.php>

<http://jan.ucc.nau.edu/lrm22/pixels2bytes/calculator.htm>

<https://toolstud.io/photo/megapixel.php?width=1920&height=1080&compare=video&calculate=compressed>

<https://kangax.github.io/compat-table/es6/> Browser compatability

<https://www.w3schools.com/js/js_versions.asp> Ecamascript versions

<https://aws.amazon.com/efs/?c=24&pt=3> EFS

<https://aws.amazon.com/rds/?c=9&pt=8> RDS

<https://aws.amazon.com/lambda/?c=7&pt=10> AWS Lambda