International Institute of Information Technology, Bangalore

Software Project Engineering Final Project

Rento - Renting Mobile App

In the Guidance of: Prof. B. Thangaraju

Teaching Assistant: Vivek Gupta

Mohit Choudhary
MT2020134

Aditya Satam
MT2020058

Table of contents

1 Abstract 3
2 Introduction 4
3 System Configuration 5
4 Software Development Life Cycle 6
    4.1 Installation 6
    4.2 Configuration Control Management GitHub 11
    4.3 CI pipeline - Jenkins 11
    4.4 Docker Architect 18
    4.5 Ansible 20
    4.6 Monitoring - ELK 22
    4.7 Build 32
    4.8 Testing 33
5 Project Overview 34
6 Use Case 39
7 Future Scaling 43
8 Conclusion 44
9 References 45

1. Abstract

Rento is a mobile app which brings you the most intuitive products search experience. If you are looking for any product on rent, our app leaves no stone unturned to offer the most relevant options for renting products.
In our project **RENTO**, user can rent any product they want, set a price on monthly basis, add some images of product and post product ad on our app. Many other user can see their product, if some user want a product they can request for a rent of that product and owner can accept whether he/she can give product to rent to that user or not. This app acts as a middle man between the two users, one who has a product and want to give it to rent, and one who want to take rent. All money matter is also handle by this app.

The architecture of our project demands three layers.

- Front end
- Middle end
- Back end

The front end of the project is handled by "Android Studio Framework" and app interface is designed by XML. The middle ware connect database which is our backend with its interface and its uses JAVA language in android studio. The backend of this project uses MONGODB as database and NodeJS to run the scripts that interact with database. The scripts are written in JavaScript.

2. Introduction

2.1 OVERVIEW

Rento is a mobile app which brings you the most intuitive products search experience. If you are looking for any product on rent, our app leaves no stone unturned to offer the most relevant options for renting products.
In our project RENTO, user can rent any product they want, set a price on monthly basis, add some images of product and post product ad on our app. Many other user can see their product, if some user want a product they can request for a rent of that product and owner can accept whether he/she can give product to rent to that user or not. This app acts as a middle man between the two users, one who has a product and want to give it to rent, and one who want to take rent. All money matter is also handle by this app.

2.2 FEATURES

- User can add their product, set their own monthly price and add multiple photos and post ad.
- Another user can see and compare multiple same type of products and compare their price and appearance and choose which one to buy on rent.
- Any user can add funds as a money in this app to make any kind of transactions.
- User can search among many items in the list to make it easy to find the desired product.

2.3 WHY DEVOPS?

Our whole approach of the project was modular, we wanted to make different sets of development modules and wanted to deploy them with every new release without any hindrance. Wanted to test the changed code with continuous integration and then continuously deploying it. So what all fills all these blanks was a culture, a philosophy DevOps. DevOps provides all the tools to increase the capability to complete above set goals within minimum time and less trouble for developers. DevOps tools consist of configuration management, test and build systems, application deployment, version control and monitoring tools. Continuous integration, continuous delivery and continuous deployment require different tools.

2.3.1 DevOps Features

- Improve deployment frequency
- Achieve faster time to market with lower failure rate
- More stable operating environments
- Improve communication and collaboration among teams

3. System Configuration

3.1 Operating System

Ubuntu 18.04.04

3.2 CPU and RAM

4 Core processor and RAM 8GB (preferable 16GB)

3.3 Language

JavaScript, XML,
Python, JAVA

3.4 Database

MongoDB

3.5 Building Tools

- Gradle to build java application
- Android studio for building app
- MongoDB for Database
- NodeJS for backend runtime environment

3.6 DevOps Tools

- Source Control Management - GitHub
- Continuous Integration - Jenkins
- Containerization - Docker
- Continuous Deployment - Ansible
- Monitoring - ELK Stack (Elasticsearch, Logstash, Kibana)

4. Software Development Life Cycle

4.1 Installation

4.1.1 Android Studio Framework

We use Android studio to build the app. We downloaded android studio 4.1.3, here are the specification of this:

```
Android Studio 4.1.3
Build #AI-201.8743.12.41.7199119, built on March 11, 2021
Runtime version: 1.8.0_242-release-1644-b01 amd64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o
Windows 10 10.0
GC: ParNew, ConcurrentMarkSweep
Memory: 1246M
Cores: 4
Registry: ide.new.welcome.screen.force=true, external.system.auto.import.disabled=true
```

Figure 1: Android Studio Framework specification

It is used as a IDE to developed a android app code, here is a reference of project structure in android studio:

Figure 2: Code structure of android studio

4.1.2 Visual Studio Code

It is used to make API (application programing interface) that connect MongoDB to our app and send and receive data from mongo db. We downloaded latest version i.e. 1.56.2 (User Setup), here is a specification of that:

Figure 3: Visiual studio code specification

There is only one file in this **app.js** which connects to MongoDB client to a URL "mongodb://localhost:27017", here is a file structure of visual studio code:

Figure 4: Code structure in VS Code

4.1.3 Installing MongoDB

We use **MongoDB** as a database. Here is the installation link to install MongoDB:
https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/
Here is our specification of MongoDB

```
Select Command Prompt
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\choud>mongo --version
MongoDB shell version v4.4.4
Build Info: {
    "version": "4.4.4",
    "gitVersion": "8db30a63db1a9d84bdcad0c83369623f708e0397",
    "modules": [],
    "allocator": "tcmalloc",
    "environment": {
        "distmod": "windows",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}
```

Figure 5: MongoDB Specification

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License.

4.1.4 Installing NodeJS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. We uses NodeJS to run the script that was written on JavaScript that connect MongoDB client to user.
To install and download NodeJS: https://nodejs.org/en/download/
Here is my specification of NodeJS:

```
Command Prompt

C:\Users\choud>npm --version
6.14.11

C:\Users\choud>
```

Figure 6: NodeJs Specification

Further details on NodeJS/npm, here is the command npm --help to show more details on npm:

```
Command Prompt

C:\Users\choud>npm --help

Usage: npm <command>

where <command> is one of:
    access, adduser, audit, bin, bugs, c, cache, ci,
    cit, clean-install, clean-install-test, completion,
    config, create, ddp, dedupe, deprecate, dist-tag,
    docs, doctor, edit, explore, fund, get, help,
    help-search, hook, i, init, install,
    install-ci-test, install-test, it, link, list, ln,
    login, logout, ls, org, outdated, owner, pack, ping,
    prefix, profile, prune, publish, rb, rebuild, repo,
    restart, root, run, run-script, s, se, search, set,
    shrinkwrap, star, stars, start, stop, t, team, test,
    token, tst, un, uninstall, unpublish, unstar, up,
    update, v, version, view, whoami

npm <command> -h  quick help on <command>
npm -l            display full usage info
npm help <term>   search for help on <term>
npm help npm      involved overview

Specify configs in the ini-formatted file:
    C:\Users\choud\.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

npm@6.14.11 C:\Program Files\nodejs\node_modules\npm

C:\Users\choud>
```

Figure 7: npm Help for more details

4.1.5 Installing JAVA

Java is the object oriented language that is the base of android app. Android app is created in JAVA language along with xml. Xml is used in creating app interface and java done all the logic building thing in the app and API calling also.
Here is the link to download and install java:
https://www.java.com/en/download/help/log_files.html

```
Command Prompt

C:\Users\choud>java --version
java 16 2021-03-16
Java(TM) SE Runtime Environment (build 16+36-2231)
Java HotSpot(TM) 64-Bit Server VM (build 16+36-2231, mixed mode, sharing)

C:\Users\choud>
```

Figure 8: java version

After downloading and installing java, we need to set the java path to environment variable so that java can be access anywhere from the system. Here is the official link we use to set on environmental variable https://java.com/en/download/help/path.html.

Figure 9: java set to environment variable

4.1.5 Installing and Setup Python

Python is a popular programming language often used to write scripts for operating systems. It is versatile enough for use in web development and app design.
We need python because of docker and ELK. Both uses python and require Python 3.6 install in the system.
We use this link to install python: https://phoenixnap.com/kb/how-to-install-python-3-ubuntu

Figure 10: Python version

4.2 SOURCE CONTROL MANAGEMENT

A Source Code Management (SCM) is a software tool used by programmers to manage the source codes. For our project every team member would clone the repository from GitHub. Create a different branch locally on their system and then merge it with master after pulling the latest code from git, resolving any conflicts and then push the changes to git.

- **git clone <repository url>** - This command copies the entire data on the git url
- **git checkout -b <branch_name>** - This command creates a new branch with the name as in 'branch_name'
- **git add <changed files>** - This command adds changes in the working directory to the staging area
- **git commit -m "message while committing"** - This command is used to save your changes to the local repository with -m used to provide a concise description that helps your teammates (and yourself) understand what happened.
- **git checkout master** - This command switches to master branch
- **git pull** - This command is used to update the local version of a repository from a remote.
- **git merge <branch_name>** - This command is used to integrate changes from another branch.
- **git push** - This command will push all the latest code to the repository.

For **Rento,** we have created 2 **GIT** repositories one for android code and one for NodeJS code. Here are the link for these repositories:

- https://github.com/MohitJain11/Renting.git
- https://github.com/MohitJain11/Renting_Node_JS.git

4.3 CI Pipeline - Jenkins

Continuous Integration(CI) is a development practice where developers integrate their code into a shared repository frequently, preferably several times a day. In this development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.

Steps for download Jenkins in ubuntu:
Step1: ~$wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
Step2: ~$sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
Step3: ~$sudo apt update
Step4: ~$sudo apt install jenkins
Step5: ~$sudo systemctl enable --now jenkins

For running Jenkins:
~$sudo systemctl start jenkins

Checking for status of Jenkins:
~$sudo systemctl start jenkins

Figure 11: Jenkins status

By default Jenkins run on 8080 port no.
Open Jenkins dashboard on URL - http://localhost:8080/
Before making new pipeline in Jenkins we need to install few required plugins in Jenkins.
Required plugins are:

- Git plugin
- Maven Integration
- Docker pipeline
- Ansible plugin

For integrating these plugins click on Manage Jenkins>Manage Plugins
and download plugins one by one.

Figure 12: Download plugins by clicking manage plugins tab in manage jenkins

After download all these plugin restart the Jenkins server. Now all the necessary plugins are installed in Jenkins server, so lets make a new pipeline in Jenkins:
For making a new pipeline, go to dashboard and click on "New Item" tab. It will open new window.

Figure 13: Click New Item to make new Pipeline in Jenkins

After clicking new item, then type the item name and click on pipeline and click ok. Thus creating a new pipeline.

Figure 14: Creating New Pipeline

Figure 15: Creating Pipeline Script

Creating a Pipeline Script, we use pipeline syntax button to generate the pipeline script. Select git and add git repository URL and select master branch. Add credential if any credential is there in your git repository and select none if repo is public and select Generate Pipeline Script, it will generate git pipeline script based on your URL and Credentials.

My Final Pipeline Script is:

```
pipeline {
    environment{
        imageName = ""
    }
    agent any

    stages {
        stage('Git pull') {
            steps {
                git branch: 'main', credentialsId: 'github-cred', url:
'https://github.com/adityasatam/spe_rent.git'
            }
        }
         stage('Clean Build') {
             steps {
                 withGradle(){
                 sh 'pwd'
                 sh 'sh ./Renting/gradlew buildEnvironment --stacktrace --scan --info'
                 }
             }
        }
        stage('Docker build to image') {
            steps {
                script{
                imageName = docker.build "sasukeaditya236/spe_major_project:latest"
                }
            }
        }
        stage('Push Docker Image') {
            steps {
                script{
                    docker.withRegistry('','docker-jenkins'){
                        imageName.push()
                    }
                }
            }
        }
        stage('Ansible Pull Docker Image') {
            steps {
                ansiblePlaybook becomeUser: null, colorized: true, disableHostKeyChecking: true,
installation: 'Ansible', inventory: 'Renting/deploy-docker/inventory', playbook: 'Renting/deploy-
docker/deploy-image.yml', sudoUser: null
            }
        }
    }
}
```

Figure 16: Pipeline Script

After creating pipeline and writing pipeline script, lets build our Jenkins server by clicking on Build Now button.

Figure 17: Pipeline Script

After project building is complete, we can see the status on build in build history.

Figure 18: Jenkins Stage View

Console output of Jenkins for GitHub after Build:

Figure 19: Console output of jenkins for GitHub

Console output of Jenkins for Gradle after Build:

Figure 20: Console output of Jenkins for Gradle after build

Console output of Jenkins for Build Docker Image & Push Docker Image:

```
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Push Docker Image)
[Pipeline] script
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withDockerRegistry
$ docker login -u sasukeaditya236 -p ******** https://index.docker.io/v1/
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/workspace/spe_android_majorproject@tmp/2543d682-99bf-40c9-9b91-d3e4
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] {
[Pipeline] isUnix
[Pipeline] sh
+ docker tag sasukeaditya236/spe_major_project:latest sasukeaditya236/spe_major_project:latest
[Pipeline] isUnix
[Pipeline] sh
+ docker push sasukeaditya236/spe_major_project:latest
The push refers to repository [docker.io/sasukeaditya236/spe_major_project]
9815417e11ff: Preparing
9534dfec042a: Preparing
```

Figure 21: Console output of Jenkins for Build Docker Image and Push Docker Image

```
aditya:8080/job/spe_android_majorproject/37/console
Dashboard > spe_android_majorproject > #37
7f03bfe4d6dc: Layer already exists
latest: digest: sha256:d6cda53b6575a041c3b8c80c3662745ca8a6d45296d7f3b9d198976f4eaaBec1 size: 1794
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Ansible Pull Docker Image)
[Pipeline] ansiblePlaybook
[spe_android_majorproject] $ /usr/bin/ansible-playbook Menting/deploy-docker/deploy-image.yml -i Menting/deploy-docker/inventory

PLAY [Pull docker image] ********************************************************

TASK [Gathering Facts] **********************************************************
[0;32mok: [localhost][0m
[0;32m[0m
TASK [Pull spe major project image] *********************************************
[0;32mok: [localhost][0m
[0;32m[0m
PLAY RECAP *********************************************************************
[0;32mlocalhost[0m                  : [0;32mok=2    [0m changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Console output of Jenkins for Ansible Pull Docker Image:


```
aditya:8080/job/spe_android_majorproject/37/console
Dashboard > spe_android_majorproject > #37
7f03bfe4d6dc: Layer already exists
latest: digest: sha256:d6cda53b6575a041c3b8c80c3662745ca8a6d45296d7f3b9d198976f4eaaBec1 size: 1794
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Ansible Pull Docker Image)
[Pipeline] ansiblePlaybook
[spe_android_majorproject] $ /usr/bin/ansible-playbook Menting/deploy-docker/deploy-image.yml -i Menting/deploy-docker/inventory

PLAY [Pull docker image] ********************************************************

TASK [Gathering Facts] **********************************************************
[0;32mok: [localhost][0m
[0;32m[0m
TASK [Pull spe major project image] *********************************************
[0;32mok: [localhost][0m
[0;32m[0m
PLAY RECAP *********************************************************************
[0;32mlocalhost[0m                  : [0;32mok=2    [0m changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```


Figure 22: Console output of jenkins for Ansible Pull Docker Image

4.4 DOCKER ARTIFACT

4.4.1 Docker Installation and Images

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications.

Docker Installation:
~$ curl -fsSL https://get.docker.com -o get-docker.sh
~$ sudo sh get-docker.sh

Some Usefull Docker Commands:
~$docker --version //checking docker version
~$docker images // checking all docker images (locally)
~$docker ps -a //checking all the containers (locally)
~$docker run hello-world //run hello-world docker image
~$sudo service docker start // to start docker service

```
aditya@aditya:~/Downloads/calculator-devops-master$ sudo apt-get install docker.io
[sudo] password for aditya:
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (19.03.8-0ubuntu1.20.04.2).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 29 not upgraded.
aditya@aditya:~/Downloads/calculator-devops-master$ docker --version
Docker version 19.03.8, build afacb8b7f0
aditya@aditya:~/Downloads/calculator-devops-master$
```

Figure 23: Docker Installation

Docker File

```
FROM openjdk:8
MAINTAINER Aditya Satam sasukeaditya236@gmail.com
COPY ./Renting/gradle-wrapper.jar ./
WORKDIR ./
CMD ["java", "-cp", "gradle-wrapper.jar", "Renting"]
```

4.4.2 Docker Hub

Docker Hub is a cloud-based repository in which Docker users and partners create, test, store and distribute container images. Through Docker Hub, a user can access public, open source image repositories, as well as use a space to create their own private repositories, automated build functions, Webhooks and work groups.

How Docker Hub repositories work:
Image repositories are spaces within Docker Hub wherein users upload and store container images. Public repositories enable users to share and collaborate on container images. Private repositories protect any sensitive or proprietary data from unauthorized persons.
To push a image in docker hub, use push command.
Eg: ~$docker push mohitjain1109/hello-world
To pull a image from docker hub, use pull command.
Eg: ~$docker pull hello-world

Creating New Repository in Docker Hub:
Open https://hub.docker.com/ and sign up to docker hub. After signup we will land up to hub dashboard, currently dashboard has no repository. For creating new repository click on Create Repository button. After that give the name of repository and description, make public or private based on your choice and click on create button, this will create the new repo and it will be visible to dashboard.
Here is my docker hub dashboard:

Figure 24: Docker Hub Project

Anybody can access this image via:
~$docker push mohitjain1109/spe_major_project:tagname

4.5 Ansible

Ansible is a open source automation platform. Automation Engine that runs ansible playbooks, playbooks are defined tasks, where we define environments and workflows.

Installation openssh-server:
~$ sudo apt install openssh-server

Lets generate rsa key required in ansible:
~$ ssh-keygen -t rsa

```
(base) mohitjain@mohitjain-HP-Notebook:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mohitjain/.ssh/id_rsa): demoKey
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in demoKey
Your public key has been saved in demoKey.pub
The key fingerprint is:
SHA256:T5RXfWyw/XowDD6+CQc1iRv2reiUyi9jd7hWASsweIo mohitjain@mohitjain-HP-Notebook
The key's randomart image is:
+---[RSA 3072]----+
|       .     o+ |
|  . +   .o o o=|
|  . o o  *o* ..o|
|   E .   .o.B.=  .|
|        S.+ +.= .|
|         o =.o + |
|           *o+ . .|
|       .+++o.o . |
|       .o*+oo    |
+----[SHA256]-----+
```

Figure 25: RSA Key for Ansible

Installation Ansible:
~$ sudo apt update
~$ sudo apt install ansible
Lets check the ansible version:
~$ansible --version

```
ansible 2.10.6 config file = /etc/ansible/ansible.cfg configured module search path =
['/home/mohitjain/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules'] ansible python
module location = /home/mohitjain/.local/lib/python3.6/site- packages/ansible executable location =
/home/mohitjain/.local/bin/ansible python version = 3.6.10 |Anaconda, Inc.| (default, May 8 2020,
02:54:21) [GCC 7.3.0]
```

Figure 26: Ansible version

Creating yml file and inventory file and added to git.

```yaml
---
- name: Pull docker image
  hosts: all
  tasks:
    - name: Pull spe major project image
      docker_image:
        name: sasukeaditya236/spe_major_project
        source: pull
```

Figure 26: Docker Image Yml

4.6 Monitoring - ELK

ELK stack makes the monitoring tool for any deployed software, it analyzes the logs and the same analysis can then be viewed on Kibana dashboard. To start with download elastic search, Logstash and Kibana from https://www.elastic.co. Run them side by side and feed the Logstash your log set after configuring it config file. In the config file we provide the details for logs. Here the config file logstash.conf Elastic search starts at localhost:9200, and Kibana starts at localhost:5601.

Commands to run:

- Elasticsearch -./path_to_elastic_search/bin/elastic
- Logstash-./path_to_logstash/bin/logstash–f./path_to_logstash.conf
- Kibana-./path_to_kibana/bin/kibana

Setting up Kibana includes creating a new index pattern under management. Add a new index pattern and press next to choose @timestamp from next window and this will create a new index pattern to view your logs.

Installations:-
LogIn/SignUp at Elastic Cloud using (https://cloud.elastic.co/login)

Figure 27: Login/SignUp to Elastic Cloud

Create a new Deployment and generate a new Cloud ID.

Figure 28: Creating New Deployment

LOGSTASH: define processor for preprocessing data using Logstash pipelines.

Figure 28: Define Processor for processing data using logstash pipelines

For Kibana:

Go to Machine Learning,
ELASTIC SEARCH: Data Visualizer to import & create Elasticsearch index to search & analysis the data.

Statistics of Log File:

elastic
Search Elastic
Machine Learning / Data Visualizer / File
File processed
Index created
Ingest pipeline created
Data uploaded
Index pattern created
Import complete
Index date
Index pattern date
Ingest pipeline date-pipeline
Documents ingested 19
View index in
Open in Data
Index Management
Index Pattern
Create Filebeat

elastic
Search Elastic
Discover
New Save Open Share Inspect
Search
KQL
Refresh
+ Add filter
date
19 hits
Search field names
Filter by type 0
type
Available fields 5
Popular
date amount name item phonenumber place type
13/May/2021:09:15:03 500 adii drum 9,399 gwalior music
13/May/2021:09:16:06 1,000 adii golf 9,399 morena sports
13/May/2021:09:17:08 200 adi guitar 8,386 udaipur sound
13/May/2021:09:19:09 5,000 mohit drilling 1,234 chandigarh tool
13/May/2021:09:20:11 100 shubham paintbrush 5,678 delhi house

Analysis:

Dashboard / Create
Download as CSV
Cancel
Save to library
Save a
Search
KQL
Last 15 minutes
Show dates
+ Add filter
Bar
phonenumber
Search field names
Field filters 0
Records
Available fields 8
amount
app
date
item
name
Median of amount
5,000
4,500
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
agra
bangalore
chandigarh
Other
Top values of place
phonenumber
Horizontal axis
Top values of place
Vertical axis
Median of amount
Drop a field or click to ad
Break down by
Drop a field or click to ad

elastic
Search Elastic
Dashboard / Create
Download as CSV
Cancel
Save to library
Save and return
Search
KQL
Last 15 minutes
Show dates
Refresh
+ Add filter
H. Stacked bar
phonenumber
Search field names
Field filters 0
item
name
phonenumber
place
type
Empty fields 0
Meta fields 3
house
music
sound
sports
tool
Top values of type
drilling
golf
guitar
drum
paintbrush
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6
Unique count of place
Top values of type
Horizontal axis
Unique count of place
Drop a field or click to add
Break down by
Top values of item
Reset layer

elastic
Search Elastic
Dashboard / Create
Download as CSV
Cancel
Save to library
Save
Search
KQL
Last 15 minutes
Show dates
+ Add filter
Line
phonenumber
Search field names
Field filters 0
Records
Available fields 8
amount
app
date
item
name
5,000
4,500
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
Median of amount
13/May/2021:09:15:03
13/May/2021:09:17:08
13/May/2021:09:20:11
Top values of date
Horizontal axis
Top values of date
Vertical axis
Median of amount
Drop a field or click to a
Break down by
Drop a field or click to a
Reset layer

elastic
Search Elastic
Dashboard / Create
Download as CSV
Cancel
Save to library
Save and return
Search
KQL
Last 15 minutes
Show dates
Refresh
+ Add filter
Pie
phonenumber
Search field names
Field filters 0
item
name
phonenumber
place
type
Empty fields 0
bangalore 24.18%
Other 24.19%
chandigarh 3.96%
gwalior 15.33%
delhi 16.38%
agra 16.36%
phonenumber
Slice by
Top values of place
Drop a field or click to add
Size by
Median of phonenumber
Reset layer

elastic
Search Elastic
Dashboard / Create
Download as CSV
Cancel
Save to library
Save and re
Search
KQL
Last 15 minutes
Show dates
Refr
+ Add filter
Treemap
phonenumber
Search field names
Field filters 0
Records
Available fields 8
amount
app
date
item
name
tool 70.08%
sports 16.82%
music 7.01%
sound 5.26%
phonenumber
Group by
Top values of type
Drop a field or click to add
Size by
Median of amount
Reset layer

ELK Stack

Start running Elasticsearch on Unix machine

```
aditya@aditya: ~/Downloads/ELK
aditya@aditya: ~/Downloads/ELK
aditya@aditya: ~/Downloads/ELK
aditya@aditya:~/Downloads/ELK$ ./elasticsearch-7.11.1/bin/elasticsearch
Java HotSpot(TM) 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed i
[2021-05-16T16:55:25,261][INFO ][o.e.n.Node               ] [aditya] version[7.11.1], pid[3645], build[default/tar/ff17057114c2
b7a/2021-02-15T13:44:09.394032Z], OS[Linux/5.8.0-53-generic/amd64], JVM[Oracle Corporation/Java HotSpot(TM) 64-Bit Server VM/11
[2021-05-16T16:55:25,266][INFO ][o.e.n.Node               ] [aditya] JVM home [/usr/lib/jvm/jdk-11.0.11], using bundled JDK [fa
[2021-05-16T16:55:25,266][INFO ][o.e.n.Node               ] [aditya] JVM arguments [-Xshare:auto, -Des.networkaddress.cache.ttl
cache.negative.ttl=10, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-Omi
Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dio.netty.allocator
4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Djava.locale.providers=SPI,COMPAT, -XX:+UseConcMarkSweepGC, -XX:CMSIn
=75, -XX:+UseCMSInitiatingOccupancyOnly, -Djava.io.tmpdir=/tmp/elasticsearch-17935829761110110928, -XX:+HeapDumpOnOutOfMemoryEr
a, -XX:ErrorFile=logs/hs_err_pid%p.log, -Xlog:gc*,gc+age=trace,safepoint:file=logs/gc.log:utctime,pid,tags:filecount=32,filesiz
4m, -XX:MaxDirectMemorySize=536870912, -Des.path.home=/home/aditya/Downloads/ELK/elasticsearch-7.11.1, -Des.path.conf=/home/adi
search-7.11.1/config, -Des.distribution.flavor=default, -Des.distribution.type=tar, -Des.bundled_jdk=true]
[2021-05-16T16:55:27,807][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [aggs-matrix-stats]
[2021-05-16T16:55:27,808][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [analysis-common]
[2021-05-16T16:55:27,808][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [constant-keyword]
[2021-05-16T16:55:27,808][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [flattened]
[2021-05-16T16:55:27,808][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [frozen-indices]
[2021-05-16T16:55:27,809][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [ingest-common]
[2021-05-16T16:55:27,809][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [ingest-geoip]
[2021-05-16T16:55:27,809][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [ingest-user-agent]
[2021-05-16T16:55:27,809][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [kibana]
[2021-05-16T16:55:27,809][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [lang-expression]
[2021-05-16T16:55:27,810][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [lang-mustache]
[2021-05-16T16:55:27,810][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [lang-painless]
[2021-05-16T16:55:27,810][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [mapper-extras]
[2021-05-16T16:55:27,810][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [mapper-version]
[2021-05-16T16:55:27,810][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [parent-join]
[2021-05-16T16:55:27,811][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [percolator]
[2021-05-16T16:55:27,811][INFO ][o.e.p.PluginsService     ] [aditya] loaded module [rank-eval]
```

Start running Kibana on Unix machine

```
aditya@aditya: ~/Downloads/ELK
aditya@aditya: ~/Downloads/ELK
aditya@aditya: ~/Downloads/ELK
aditya@aditya:~/Downloads/ELK$ ./kibana-7.11.1-linux-x86_64/bin/kibana
  log   [16:55:46.735] [info][plugins-service] Plugin "visTypeXy" is disabled.
  log   [16:55:46.816] [warning][config][deprecation] Config key [monitoring.cluster_alerts.email_notifications.email_address] will be required for email not
  log   [16:55:47.017] [info][plugins-system] Setting up [101] plugins: [taskManager,licensing,globalSearch,globalSearchProviders,code,usageCollection,xpackLe
emetryCollectionXpack,kibanaUsageCollection,kibanaLegacy,securityOss,newsfeed,mapsLegacy,translations,share,legacyExport,expressions,charts,embeddable,uiActio
ability,console,consoleExtensions,apmOss,searchprofiler,painlessLab,grokdebugger,management,indexPatternManagement,advancedSettings,fileUpload,savedObjects,vi
eTimelion,features,licenseManagement,dataEnhanced,watcher,canvas,visTypeTimeseries,visTypeTimeseriesEnhanced,visTypeTagcloud,visTypeTable,visTypeMarkdown,visT
nputControlVis,graph,timelion,dashboard,dashboardEnhanced,visualize,discover,discoverEnhanced,savedObjectsManagement,spaces,security,savedObjectsTagging,maps,
napshotRestore,enterpriseSearch,nl,encryptedSavedObjects,fleet,indexManagement,rollup,remoteClusters,crossClusterReplication,indexLifecycleManagement,dashboa
es,eventLog,actions,alerts,triggersActionsUi,stackAlerts,securitySolution,case,infra,monitoring,logstash,apm,uptime]
  log   [16:55:47.019] [info][plugins][taskManager] TaskManager is identified by the Kibana UUID: fda37ce1-6543-48c9-a771-7e6d8a0ea3b4
  log   [16:55:47.310] [warning][config][plugins][security] Generating a random key for xpack.security.encryptionKey. To prevent sessions from being invalida
yptionKey in the kibana.yml or use the bin/kibana-encryption-keys command.
  log   [16:55:47.311] [warning][config][plugins][security] Session cookies will be transmitted over insecure connections. This is not recommended.
  log   [16:55:47.355] [warning][config][plugins][reporting] Generating a random key for xpack.reporting.encryptionKey. To prevent sessions from being invali
ncryptionKey in the kibana.yml or use the bin/kibana-encryption-keys command.
  log   [16:55:47.357] [info][config][plugins][reporting] Chromium sandbox provides an additional layer of protection, and is supported for Linux Ubuntu 20.0
  log   [16:55:47.391] [warning][config][encryptedSavedObjects][plugins] Generating a random key for xpack.encryptedSavedObjects.encryptionKey. To decrypt em
, please set xpack.encryptedSavedObjects.encryptionKey in the kibana.yml or use the bin/kibana-encryption-keys command.
  log   [16:55:47.395] [warning][fleet][plugins] Fleet APIs are disabled because the Encrypted Saved Objects plugin uses an ephemeral encryption key. Please s
in the kibana.yml or use the bin/kibana-encryption-keys command.
  log   [16:55:47.422] [warning][actions][actions][plugins] APIs are disabled because the Encrypted Saved Objects plugin uses an ephemeral encryption key. Pl
nKey in the kibana.yml or use the bin/kibana-encryption-keys command.
  log   [16:55:47.431] [warning][alerting][alerts][plugins][plugins] APIs are disabled because the Encrypted Saved Objects plugin uses an ephemeral encryptio
encryptionKey in the kibana.yml or use the bin/kibana-encryption-keys command.
  log   [16:55:47.499] [info][monitoring][monitoring][plugins] config sourced from: production cluster
  log   [16:55:47.683] [info][savedobjects-service] Waiting until all Elasticsearch nodes are compatible with Kibana before starting saved objects migrations.
  log   [16:55:47.872] [info][savedobjects-service] Starting saved objects migrations
  log   [16:55:48.102] [info][plugins-system] Starting [101] plugins: [taskManager,licensing,globalSearch,globalSearchProviders,code,usageCollection,xpackLeg
etryCollectionXpack,kibanaUsageCollection,kibanaLegacy,securityOss,newsfeed,mapsLegacy,translations,share,legacyExport,expressions,charts,embeddable,uiActions
lity,console,consoleExtensions,apmOss,searchprofiler,painlessLab,grokdebugger,management,indexPatternManagement,advancedSettings,fileUpload,savedObjects,visT
imelion,features,licenseManagement,dataEnhanced,watcher,canvas,visTypeTimeseries,visTypeTimeseriesEnhanced,visTypeTagcloud,visTypeTable,visTypeMarkdown,visTyp
utControlVis,graph,timelion,dashboard,dashboardEnhanced,visualize,discover,discoverEnhanced,savedObjectsManagement,spaces,security,savedObjectsTagging,maps,lo
pshotRestore,enterpriseSearch,nl,encryptedSavedObjects,fleet,indexManagement,rollup,remoteClusters,crossClusterReplication,indexLifecycleManagement,dashboard
,eventLog,actions,alerts,triggersActionsUi,stackAlerts,securitySolution,case,infra,monitoring,logstash,apm,uptime]
  log   [16:55:48.821] [info][plugins][watcher] Your basic license does not support watcher. Please upgrade your license.
  log   [16:55:48.825] [info][crossClusterReplication][plugins] Your basic license does not support crossClusterReplication. Please upgrade your license.
  log   [16:55:48.825] [info][kibana-monitoring][monitoring][monitoring][plugins] Starting monitoring stats collection
  log   [16:55:49.681] [info][listening] Server running at http://localhost:5601
  log   [16:55:50.327] [info][server][Kibana][http] http server running at http://localhost:5601
```

Start running logstash using config file to process the log file

```
aditya@aditya: ~/Downloads/ELK
aditya@aditya:~/Downloads/ELK$ ./logstash-7.11.1/bin/logstash -f ELK_files/renting_logstash.conf
Using JAVA_HOME defined java: /usr/lib/jvm/jdk-11.0.11
WARNING, using JAVA_HOME while Logstash distribution comes with a bundled JDK
Java HotSpot(TM) 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in
Sending Logstash logs to /home/aditya/Downloads/ELK/logstash-7.11.1/logs which is now configured via log4j2.properties
[2021-05-16T21:38:47,776][INFO ][logstash.runner          ] Starting Logstash {"logstash.version"=>"7.11.1", "jruby.version"=>"j
t(TM) 64-Bit Server VM 11.0.11+9-LTS-194 on 11.0.11+9-LTS-194 +indy +jit [linux-x86_64]"}
[2021-05-16T21:38:48,594][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or command
[2021-05-16T21:38:51,362][INFO ][org.reflections.Reflections] Reflections took 82 ms to scan 1 urls, producing 23 keys and 47 va
[2021-05-16T21:38:53,305][INFO ][logstash.outputs.elasticsearch][main] Elasticsearch pool URLs updated {:changes=>{:removed=>[],
[2021-05-16T21:38:53,625][WARN ][logstash.outputs.elasticsearch][main] Restored connection to ES instance {:url=>"http://localho
[2021-05-16T21:38:53,768][INFO ][logstash.outputs.elasticsearch][main] ES Output version determined {:es_version=>7}
[2021-05-16T21:38:53,777][WARN ][logstash.outputs.elasticsearch][main] Detected a 6.x and above cluster: the `type` event field
ion=>7}
[2021-05-16T21:38:53,999][INFO ][logstash.outputs.elasticsearch][main] New Elasticsearch output {:class=>"LogStash::Outputs::Ela
[2021-05-16T21:38:54,167][INFO ][logstash.outputs.elasticsearch][main] Using a default mapping template {:es_version=>7, :ecs_co
[2021-05-16T21:38:54,603][INFO ][logstash.outputs.elasticsearch][main] Attempting to install template {:manage_template=>{"index
>{"index.refresh_interval"=>"5s", "number_of_shards"=>1}, "mappings"=>{"dynamic_templates"=>[{"message_field"=>{"path_match"=>"
ype"=>"text", "norms"=>false}}}, {"string_fields"=>{"match"=>"*", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "n
 "ignore_above"=>256}}}}}], "properties"=>{"@timestamp"=>{"type"=>"date"}, "@version"=>{"type"=>"keyword"}, "geoip"=>{"dynamic"=
>{"type"=>"geo_point"}, "latitude"=>{"type"=>"half_float"}, "longitude"=>{"type"=>"half_float"}}}}}}
[2021-05-16T21:38:54,741][INFO ][logstash.javapipeline    ][main] Starting pipeline {:pipeline_id=>"main", "pipeline.workers"=>4
0, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/home/aditya/Downloads/ELK/ELK_files/renting_logstash.conf"], :thread=>"#
[2021-05-16T21:38:56,300][INFO ][logstash.javapipeline    ][main] Pipeline Java execution initialization time {"seconds"=>1.55}
[2021-05-16T21:38:56,681][INFO ][logstash.inputs.file     ][main] No sincedb_path set, generating one based on the "path" settin
7.11.1/data/plugins/inputs/file/.sincedb_fff94a79623609b9bead38c6ff78eaa7", :path=>["/home/aditya/Downloads/ELK/ELK_files/rentin
[2021-05-16T21:38:56,722][INFO ][logstash.javapipeline    ][main] Pipeline started {"pipeline.id"=>"main"}
[2021-05-16T21:38:56,873][INFO ][logstash.agent           ] Pipelines running {:count=>1, :running_pipelines=>[:main], :non_runn
[2021-05-16T21:38:56,962][INFO ][filewatch.observingtail  ][main][b62e51032b0929dc69214c9125779c71d1a204ba9e69aa3cc8d3673e70bd06
cedb collections
[2021-05-16T21:38:57,368][INFO ][logstash.agent           ] Successfully started Logstash API endpoint {:port=>9600}
{
      "@version" => "1",
    "@timestamp" => 2021-05-16T16:08:57.696Z,
          "host" => "aditya",
```

Grok filtering and parsing the log file

```
aditya@aditya: ~/Downloads/ELK
}
{
        "line4" => "mumbai",
     "@version" => "1",
        "line2" => "guitar",
         "host" => "aditya",
         "path" => "/home/aditya/Downloads/ELK/ELK_files/renting.log",
       "thread" => "Renting.java",
       "logger" => "renting.Renting",
      "message" => "13/May/2021:09:46:11 215 [Renting.java] [INFO] renting.Renting [rent] ; 8380 ; dilip ; guitar ; sound ; mumbai ; 250",
        "line3" => "sound",
        "line6" => "250",
         "line" => "8380",
        "line1" => "dilip",
        "level" => "INFO",
   "@timestamp" => 2021-05-13T04:16:11.215Z,
       "action" => "rent"
}
{
        "line4" => "delhi",
     "@version" => "1",
        "line2" => "paintbrush",
         "host" => "aditya",
         "path" => "/home/aditya/Downloads/ELK/ELK_files/renting.log",
       "thread" => "Renting.java",
       "logger" => "renting.Renting",
      "message" => "13/May/2021:09:53:25 215 [Renting.java] [INFO] renting.Renting [rent] ; 5673 ; raju ; paintbrush ; house ; delhi ; 20",
        "line3" => "house",
        "line6" => "20",
         "line" => "5673",
        "line1" => "raju",
        "level" => "INFO",
   "@timestamp" => 2021-05-13T04:23:25.215Z,
       "action" => "rent"
}
```

renting_logstash.conf file

```
input {
  file {
    path => "/home/aditya/Downloads/ELK/ELK_files/renting.log"
    start_position => "beginning"
  }
}

filter {
  grok {
    match => [
      "message", "%{HTTPDATE:timestamp_string} \[%{GREEDYDATA:thread}\] \[%{LOGLEVEL:level}\] %
{GREEDYDATA:logger} \[%{GREEDYDATA:action}\] \; %{GREEDYDATA:line} \; %{GREEDYDATA:line1} \; %
{GREEDYDATA:line2} \; %{GREEDYDATA:line3} \; %{GREEDYDATA:line4} \; %{GREEDYDATA:line6}"
    ]
  }

  date {
    match => ["timestamp_string", "dd/MMM/YYYY:HH:mm:ss SSS"]
  }

  mutate {
    remove_field => [timestamp_string]
  }
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "renting_elastic"
  }

  stdout {
    codec => rubydebug
  }
}
```

Open Kibana using http://localhost:5601 & Find the new index been created using logstash automatically.

Visualize the data using Machine Learning > Data Visualizer

Visualize > Lens to create & analyze using different graphs

4.7 BUILD

4.3.1 GRADLE

Gradle is a build system (open source) which is used to automate building, testing, deployment etc. “Build. gradle” are scripts where one can automate the tasks. For example, the simple task to copy some files from one directory to another can be performed by Gradle build script before the actual build process happens.
We have Gradle scripts folder in our project structure which contains two gradle files, one is app level gradle file and another one is project level gradle file.

Figure : Gradle structure

Here are the links of these gradle:

- https://github.com/MohitJain11/Renting/blob/master/build.gradle
- https://github.com/MohitJain11/Renting/blob/master/app/build.gradle

4.8 TEST

Jenkins provides us with continuous integration which includes integrated testing, so every time we push code to github it integrates all the different modules of the project and tests their proper functioning.

For testing within java code (Rento) we use AndroidJunit. AndroidJunittest cases are written in covering the scope of the project. Also AndroidJunit dependency is added in Gradle file for Gradle to resolve the dependency. While we build our project using Gradle, Gradle runs all the defined test cases. If successful then only moves to building it, otherwise the entire process fails.

```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'com.google.android.material:material:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'androidx.navigation:navigation-fragment:2.2.2'
    implementation 'androidx.navigation:navigation-ui:2.2.2'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
    implementation 'com.squareup.retrofit2:retrofit:2.5.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.5.0'
    // Unit testing dependencies
    testImplementation 'junit:junit:4.12'
}
```

Figure : Dependency Used

5. Project Overview

5.1 Login Page

When we start the application we see the login activity.

Fig 30: LogIn Page

5.2 SignUp Page

User can registered himself/herself through signup app.

Fig 31: SignUp Page

5.3 Dashboard Page

Here we can see all the products and search for a particular product user want.

Fig 31: Dashboard Page

5.4 Ad Product Page

We can add product from this page, here user can fill product name, description, price and location of the product and upload image and post the ad.

Fig 32: Ad Product Page

Fig 33: Upload Image Page

5.5 Product Detail Page

User can see a particular product and their details before request of rent.

Fig 34: Product Detail Page

5.6 Menu Page

All the menu this app can do.

Fig 35: Menu Page

5.7 Notification Page

User can see which user request for which product in notification tab.

Fig 35: Menu Page

5.8 My Ads Page

User can see their ads in this page.

Fig 35: My Ads Page

5.8 app.js

This files contains the API that connects MongoDB and android app.

```
const express = require('express')
const app = express()
const mongoClient = require('mongodb').MongoClient
const url = "mongodb://localhost:27017"
app.use(express.json())

mongoClient.connect(url, (err, db)=>{
    if(err) {
        console.log("Error while connecting mongo client")
    }else {
        const myDb = db.db('RentingDB')
        const collection = myDb.collection('User')

        app.post('/signup', (req, res)=>{
            const newUser = {
                user_name: req.body.user_name,
                user_mobile: req.body.user_mobile,
                user_password: req.body.user_password,
                user_fund: 0
            }
            const query = {user_mobile: newUser.user_mobile}
            collection.findOne(query, (err, result)=>{
                if(result == null){
                    collection.insertOne(newUser, (err, result)=>{
                        res.status(200).send()
                    })
                } else {
                    res.status(400).send()
                }
            })
        })
```

Fig 35: Code snippit of app.js

```
C:\Users\choud\Desktop\NodeJs Project\Renting>node app.js
(node:18408) [MONGODB DRIVER] Warning: Current Server Discovery and Monitoring engine is deprecated,
and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass
option { useUnifiedTopology: true } to the MongoClient constructor.
(Use `node --trace-warnings ...` to show where the warning was created)
Listening on port 3000...
```

Fig 35: server running

6. Use Case

App launch
Have an account?
Yes
Remember password?
No
Yes
No
Login
Forget Password
Sign Up
No
Login Success?
Reset password?
No
Sign Up Success?
No
Yes
Yes
Yes
Dashboard

Dashboard
View Item
Add Item
View/Add Fund
Menu
Yes
Added successfully?
No
Item Details
Choose options: debit, credit, netbanking
Reset password / View/Edit Profile / Track Orders / Order Notifications
Yes
Added successfully?
Rent or Buy?
No
Rent
Buy
Error Window/Retry
Confirm Rent Request?
No
Yes
Confirm Buy Request?
Yes
No
Send Notification to Buyer to pay the amount
Send Notification to Buyer that request denied
Dashboard

Send Notification to Buyer to pay the amount
Pay Amount?
No
Yes
Send Notification to Seller request denied
Order Success

Dashboard
Menu
Track Order/
Order History
User Profile
Notification
Forgot Password/
Reset Password

7. Future Scaling

We would like to add more features like what if someone like the rented product and want to keep it so add a buying option. Also what if someone doesn't like the rented product and want to return before rent time so add another option.

We want to switch to real money in place of our own money(funds) by integrating online payments system that support online money transfers.

We need to use dashboard to analyze & visualize the user information more to handle growing no. of users by improving software and adding hardware.

8. Conclusion

We have successfully build a mobile app that can post and rent a product. User can use this platform and earn money by renting their products and other users can easily find the required product for rent so there is no need to buy them expensively. This app uses many tools such as: GitHub, Gradle, Docker, Ansible. These tools are integrated using Jenkins. The entire pipeline has been automated. For deployment of the entire project.

For deployment of the entire project , it takes a minimum of a minute and maximum of 15 minutes time.

The DevOps methodology and lifecycle tools prove to be better than the Agile methodology in terms of technical, cultural and business benefits. By minimizing friction between independent teams, DevOps enables a collaborative approach for enterprise software development and delivery that reflects the needs of the entire application lifecycle for today's modern enterprises.

Thus, we can develop, test, deploy and monitor the application easily.

9. References

[1] https://www.wikipedia.org/

[2] https://material-ui.com/

[3] https://www.docker.com/

[4] https://in.pinterest.com

[5] https://stackoverflow.com/

[6] https://developer.android.com/training/basics/firstapp

[7] https://www.tutorialspoint.com/mongodb/index.htm