

# Godam

(Goods Availability Management)

---

Authored by:

Prashant Sangtani (MT2018082)

Rajeev Pankaj Shukla (MT2018091)

Shubham Darokar (MT20180113)



---

# Contents

Abstract .....	3
Introduction .....	4
Why DevOps? .....	4
About the application .....	8
Software Development Life Cycle .....	9
Project Scope .....	9
Project Architecture .....	10
Source Control Management .....	12
Git .....	12
Build Tools .....	13
Gradle .....	14
Test Tools .....	15
JUnit .....	15
Continuous Integration and Deployment Tools .....	16
Jenkins .....	16
Monitoring Tools .....	18
ELK Stack .....	18
Results .....	20
Future Work .....	21
Conclusion .....	21
List of Figures .....	23
References .....	24

---

# Abstract

Smartphones are mobile devices that travel with their owners and with costs of smartphones taking a dive and power inside them increasing, smartphones have become a great tool to provide increasingly powerful services. Taking advantage of portability and ease of use, we are introducing “Godam” an android application software which helps for the businesses to operate warehouses, where owner can keep the records of available stock.

---

# Introduction

## Why DevOps?

DevOps is a software development approach that focuses on the collaboration between developers and operations, where developers are empowered to own their code from cradle to grave and operations develops tools for automation to be used by their developers. Using a DevOps model, we will share a common goal to quickly deliver quality products and services through more frequent deployment and collaboration.



*Figure 1 : DevOps Culture*

---

DevOps culture believes in the concept of automation to optimize productivity and minimize human errors.

It provides solutions to two main perspectives in the following way:

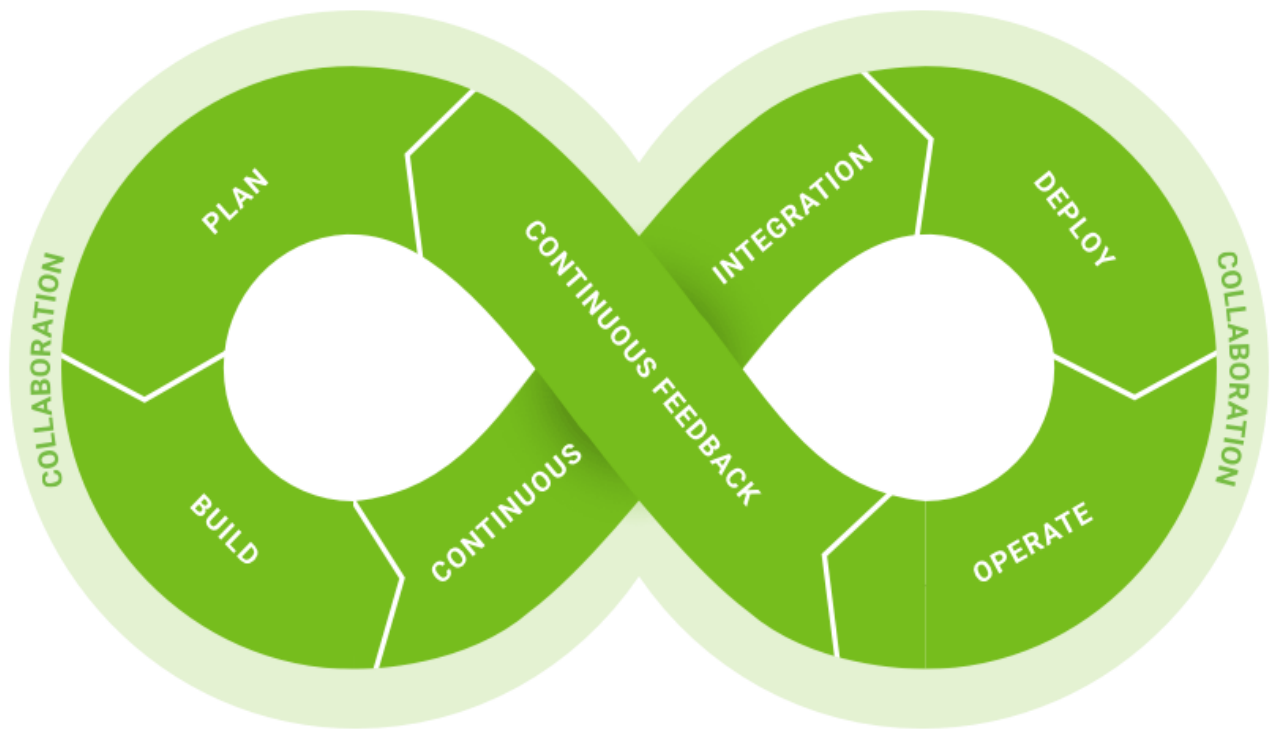
### 1. FROM THE TEAM PERSPECTIVE

- The build and deployment related tasks are always painful in a non-DevOps environment.
- These are the stages where specific environment related issues surge and where all the development that has been implemented by many team members become merged to produce the final version that is going to be delivered.
- These kinds of issues are common in a non-DevOps environment since the DEV and IT teams work separately, with everyone just taking care of his/her own individual responsibilities. With this sort of team organization, people can spend hours and even days trying to reproduce and resolve every emerging problem, something that could end up with frustrated team members, system instability, and delays on deliveries.
- The DevOps culture overcomes these issues by assigning the delivery responsibilities to the entire team and by promoting collaboration and automation throughout the delivery process to minimize pre and post deployment headaches.

DevOps relies on two main automated processes:

- continuous integration
- continuous delivery

both of these are implemented to get faster development cycles without compromising the product's quality. Due to their trusted automated process, DevOps also cares about deployment in diverse environments. In this way, the development team gains more time to concentrate its efforts on new customer necessities rather than on researching deployment related issues. Additionally, due to the shorter development cycles, problem solving tends to be simpler and detected in earlier stages.



*Figure 2 : DevOps Cycle*

## 2. FROM THE CUSTOMERS PERSPECTIVE

- Customers' business can experience negative impacts due to production issues not detected during the development process. The resulting implications can be a system malfunction, wrong accounting calculations, or even system inaccessibility.
- All those aspects could result in negative side effects like user dissatisfaction and a decrease in company revenues that could lead to the loss of millions of dollars.
- Innovation is also a key factor that impacts customers' business effectiveness. The more quickly new features and bugs fixes are released, the faster the adaptation is to changing markets. Delays in this aspect can result in the loss of an opportunity window for a business.
- DevOps improves the customer's business value by providing a continuous delivery of products that satisfy their needs. It also promotes faster delivery of features due to the automation process implemented all over the supply chain.

- 
- Furthermore, system recovery time is minimized due to the shorter delivery cycles and the improved communication and collaboration between developers and operation team members, which is combined with the automated delivery process that verifies that each change is functional and safe to release.
  - As a consequence, time and money can be destined to system innovation rather than to fixing bugs or maintenance related tasks, which allows businesses to grow and compete more efficiently in the market.

---

## About the application

“Godam” is the acronym for Goods Availability Management and is derived from Hindi word “गोदाम” which means Warehouse. “Godam” can be used in small stores for keeping track of stock easily and save the time.

### INTRODUCTION:

- Godam is software which is helpful for the businesses to operate warehouses, where owner keeps the records of available stocks.
- This project eliminates the paperwork, human faults, manual delay and speed up the process.
- Godam will have the ability to track available stock.
- This is simple, fast and intelligent stock management that can be used by anyone who has a smartphone.

### EXISTING SYSTEM:

- Manual calculation of stock present in any store is time-consuming and very risky.
- The workers cannot maintain the store when there is no owner.

### PROPOSED SYSTEM:

The proposed system is an Android application, it is better than existing system because of following reasons:

- Easily accessible by anyone.
- Overall stock of products can be viewed by owner.
- It doesn't require manpower, so it is not expensive.

### ADVANTAGES:

- This Godam project will be a great help for the stores because it is a great difficult task that to manage or get stock information in warehouse and stores. It can also manage the stock by doing operations (add, remove) so that the person will be notified when to get the new stock into the store.

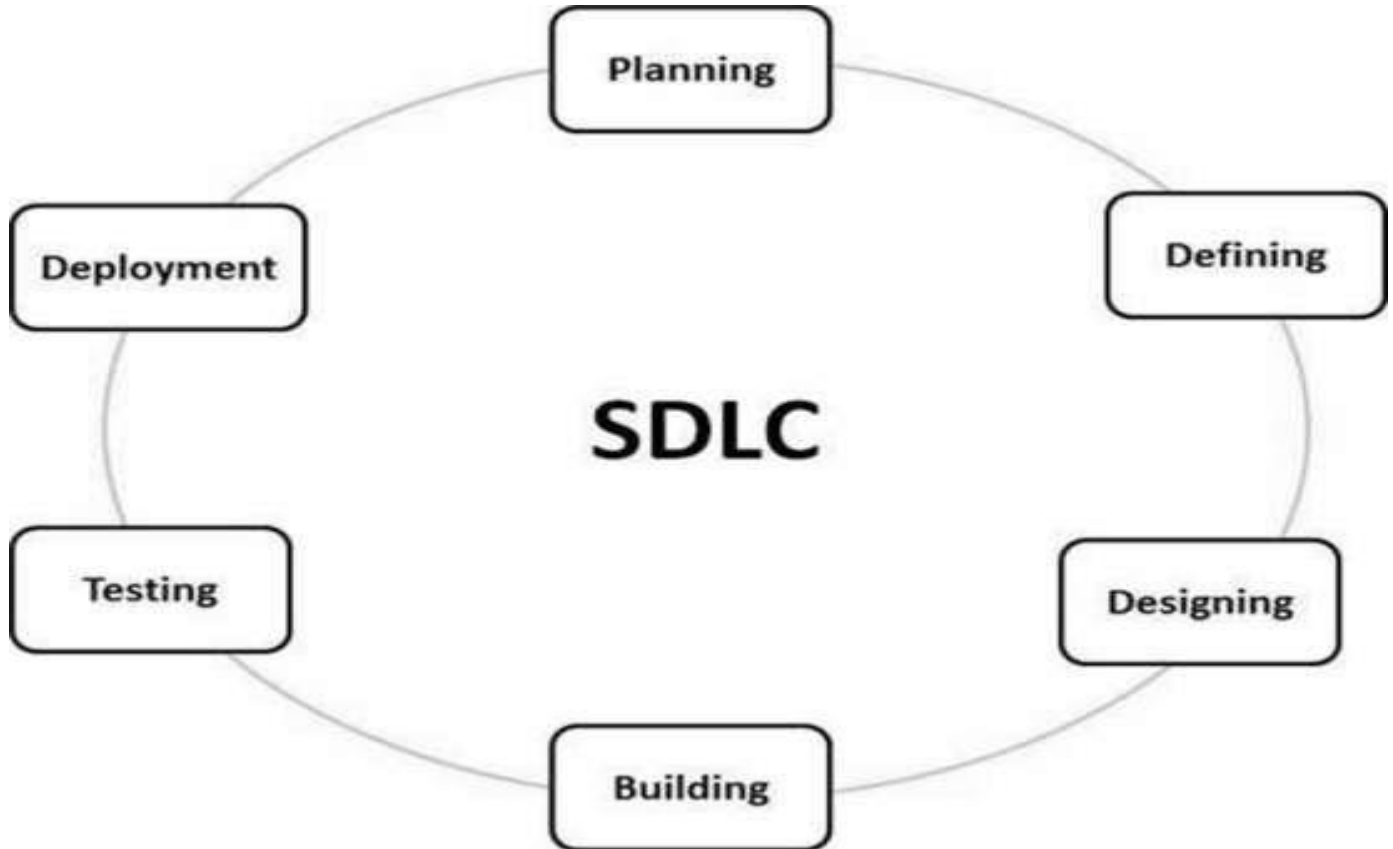


---

# Software Development Life Cycle

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



*Figure 3 : Software Development Life Cycle*

## Project Scope

- This android application can be used for tracking the records of available stocks.
- An owner can get availability of a product without being present at warehouse.
- This android application will provide the ease of updating stocks details (Id, Quantity etc.) of different products by inventory manager as and when needed.
- An owner can track details of products in his warehouse in real time.
- An owner has capability to assign roles as Inventory Manager (who updates stock details) and Supervisor (who supervises work of different Inventory Managers).

## Project Architecture

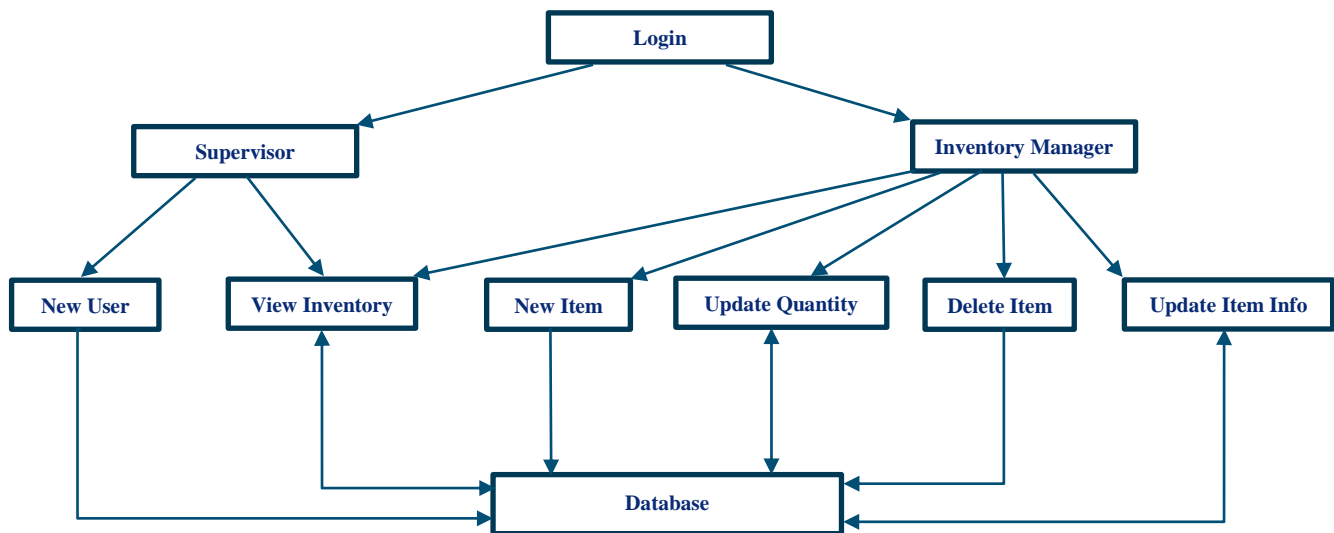
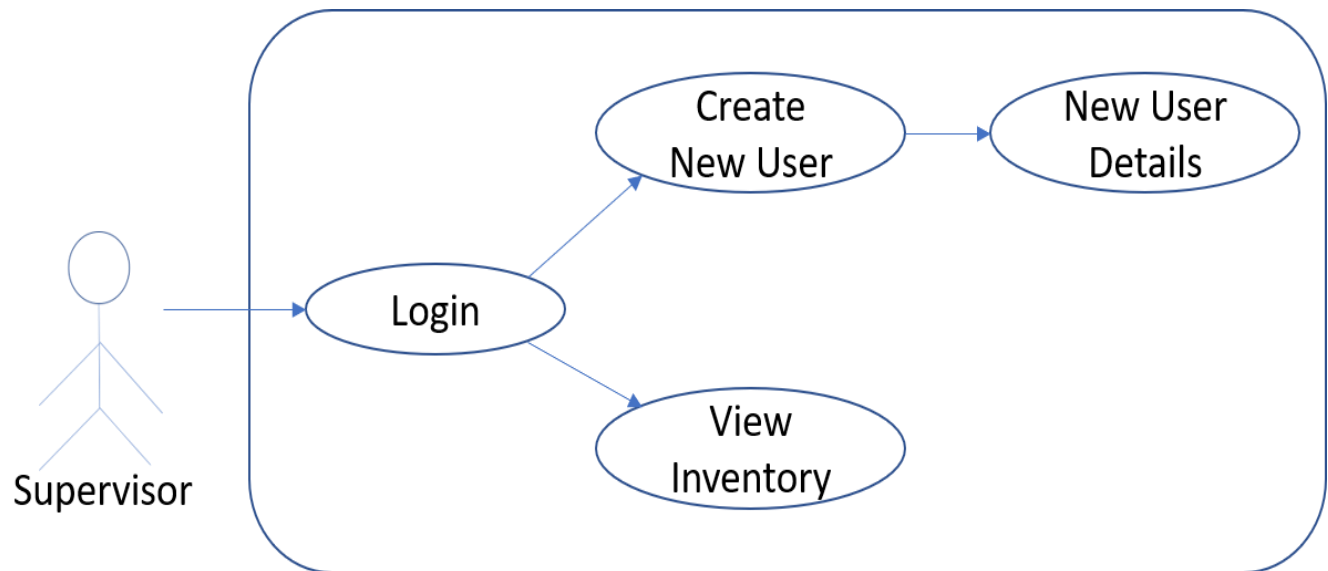
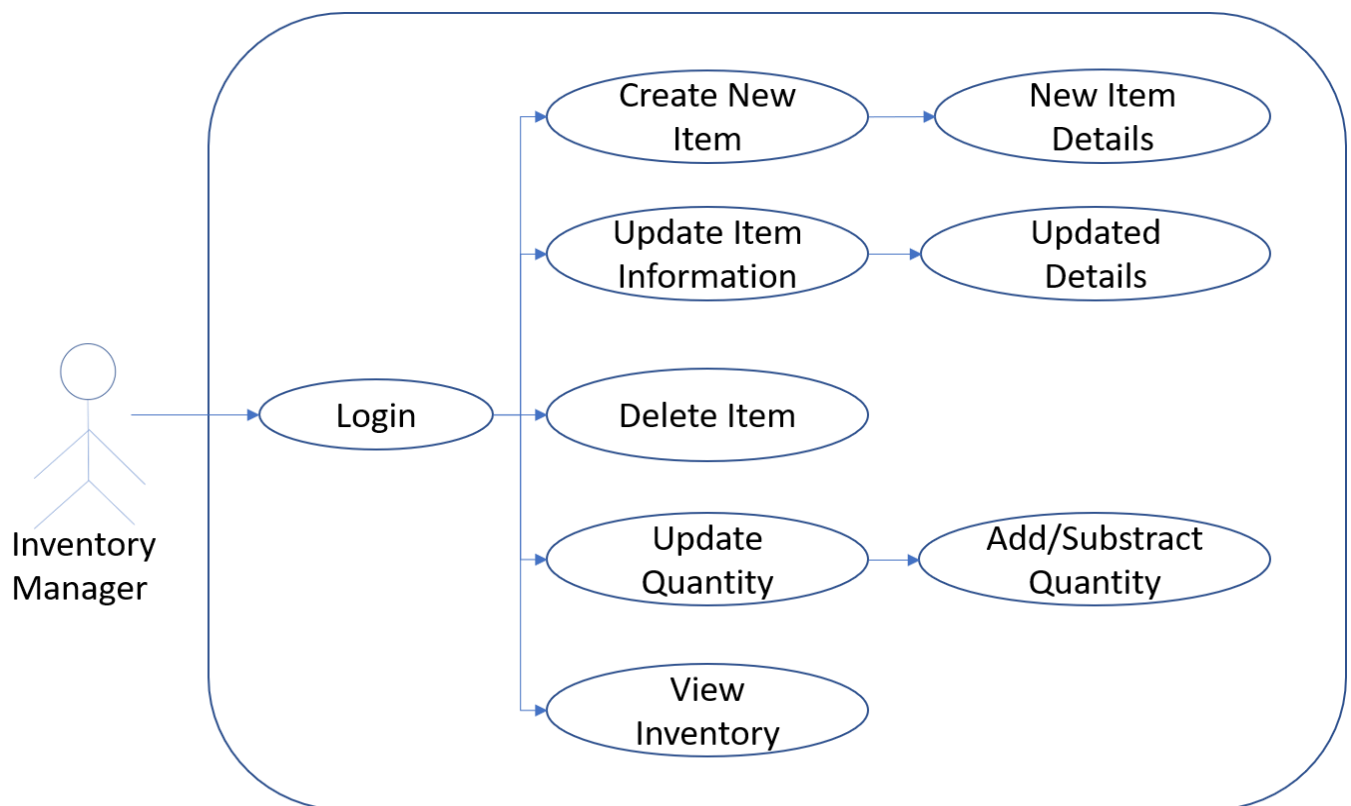


Figure 4: Godam Workflow

1. Login: It is home page of our application software, we use Email-Id and password to authenticate user and the same are used to authorize them as Supervisor or Inventory Manager.
2. Supervisor: A Supervisor has authorization rights to create another Supervisor or an Inventory Manager and view available stocks in the warehouse.
3. Inventory Manager: An Inventory manager has rights to create a new item, update quantity of already existing item, delete an already existing item or update information of the same.
4. New User: A new user can only be created by a Supervisor. A new can either be a Supervisor or an Inventory Manager.
5. View Inventory: It shows real time stock information of products filtered by category and brand.
6. New Item: Only an inventory manager can create a new item. It takes input from inventory manager and update the database with new record.
7. Update Quantity: An inventory manager can update quantity of an existing item using this activity.
8. Delete Item: An inventory manager can delete a particular item from database.
9. Update Item Info: Inventory manager can update/correct information of existing item in the database.



*Figure 4 : Use Case Diagram for Supervisor*



*Figure 5 : Use Case Diagram for Inventory Manager*

---

## Source Control Management

Source control (or version control) is the practice of tracking and managing changes to code. Source control management (SCM) systems provide a running history of code development and help to resolve conflicts when merging contributions from multiple sources.

In our project we have used ‘**Git**’ as SCM tool.

### Git

Git is an open-source distributed source code management system. Git allows us to create a copy of our repository known as a branch. Using this branch, we can then work on our code independently from the stable version of our codebase. Once we are ready with our changes, we can store them as a set of differences, known as a commit. We can pull in commits from other contributors to our repository, push our commits to others, and merge our commits back into the main version of the repository.

We created a repository named “[godam-warehouse-inventory-management](https://github.com/shubh1410/godam-warehouse-inventory-management)” in GitHub. Firstly, we created a base version of project and pushed it to our repository. Then, for every feature we created a new branch from the current version so that contributor can commit and push changes for that feature. After all the branches are updated by the contributors, they are merged by an integrator resolving conflicts if any.

Git    Repository    Link:    <https://github.com/shubh1410/godam-warehouse-inventory-management-.git>

77 commits
3 branches
25 releases
3 contributors

Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download

RajeevPankajShukla added ViewInventory functionality
Latest commit f57412b 10 hours ago

.idea	logo must be changed	11 hours ago
app	added ViewInventory functionality	10 hours ago
gradle/wrapper	initial commit	26 days ago
.gitignore	initial commit	26 days ago
README.md	Update README.md	20 days ago
build.gradle	added firebase authentication:MainActivity.java	24 days ago
gradle.properties	initial commit	26 days ago
gradlew	initial commit	26 days ago
gradlew.bat	initial commit	26 days ago
settings.gradle	initial commit	26 days ago

**Figure 6 : Godam Git Repository**

Overview
Yours
Active
Stale
All branches
Search branches...

Default branch

master Updated 10 hours ago by Rajeev
Default

Active branches

viewInventoryIssues	Updated 2 days ago by Prashantsangtani	5   0	New pull request	
db_changes	Updated 4 days ago by Prashantsangtani	20   0	New pull request	

**Figure 7 : Godam Git Repository Branches**

## Build Tools

Build automation is the act of scripting or automating a wide variety of tasks that software developers do in their day-to-day activities including things like:

- compiling computer source code into binary code
- packaging binary code
- running tests
- deployment to production systems
- creating documentation and/or release notes

In our project, we have used ‘**Gradle**’ as a build tool.

## Gradle

Gradle is an open-source build automation tool focused on flexibility and performance. Gradle build scripts are written using a Groovy or Kotlin DSL.

Gradle is an open-source build automation system that builds upon the concepts of Apache Ant and Apache Maven.

```
apply plugin: 'com.android.application'
android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.godam"
        minSdkVersion 17
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    allprojects {
        repositories {
            jcenter()
            google()
        }
    }
}
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0-alpha3'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.android.support:support-media-compat:28.0.0'
    implementation 'com.android.support:support-v4:28.0.0'
    implementation 'com.google.firebase:firebase-core:16.0.1'
    implementation 'com.google.firebase:firebase-database:16.0.1'
    implementation 'com.google.firebase:firebase-auth:16.0.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.android.support:cardview-v7:28.0.0-alpha3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'
    androidTestImplementation 'com.android.support.test:rules:1.0.2'
    androidTestImplementation 'androidx.test:rules:1.1.1'
    androidTestImplementation 'androidx.test.runner:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-intents:3.1.1'
    androidTestImplementation 'androidx.test:core:1.0.0'
    androidTestImplementation 'androidx.test.ext:junit:1.1.0'
    implementation 'com.android.support:recyclerview-v7:28.0.0-alpha3'
}
apply plugin: 'com.google.gms.google-services'
```

Figure 8 : Gradle Dependencies

## Test Tools

Testing is performed by any organization to ensure the quality of the product and continual improvement on the compliance. Testing is a process of executing a program with intent of finding an error or testing is a process used to help to identify the correctness, completeness and quality of developed computer software.

In our project, we have used '**JUnit**' testing framework to test our application software.

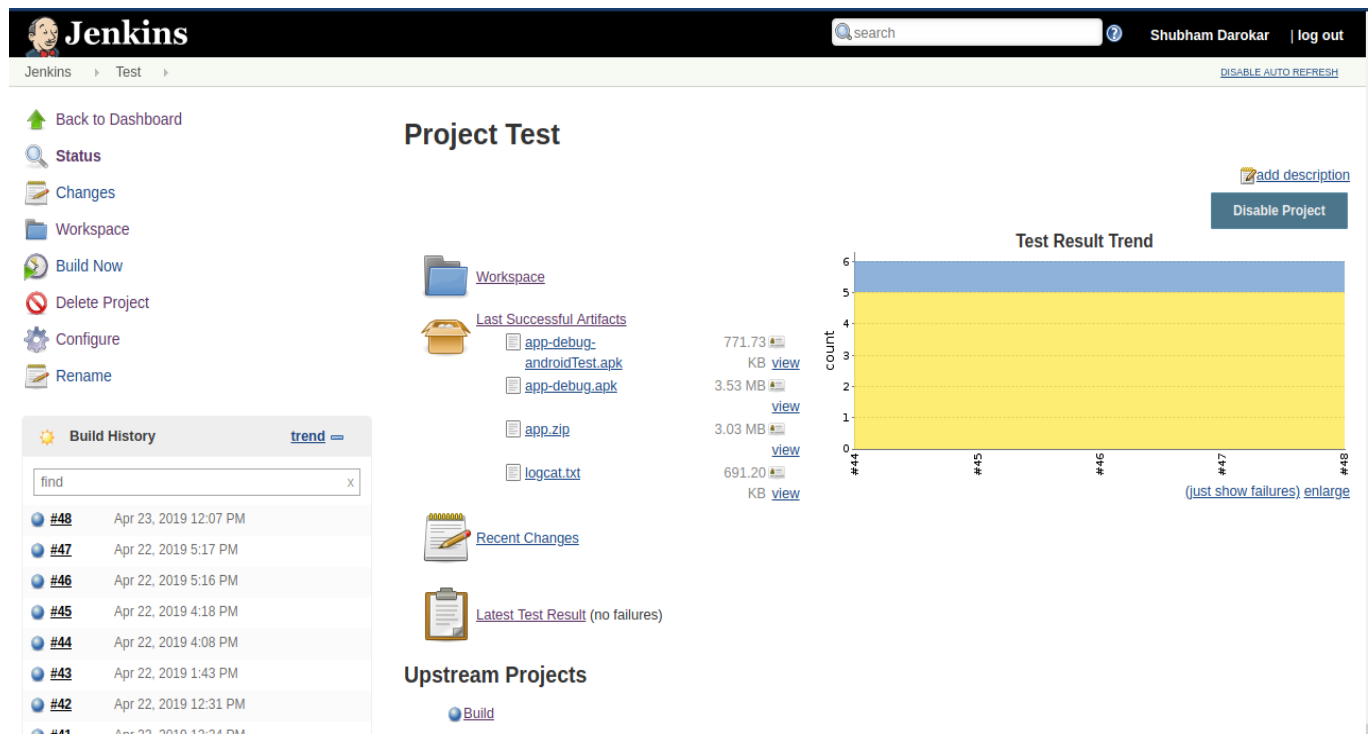
## JUnit

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks which is collectively known as xUnit that originated with SUnit.

**/\*To be filled by Shubham\*/**

Test Case 1:

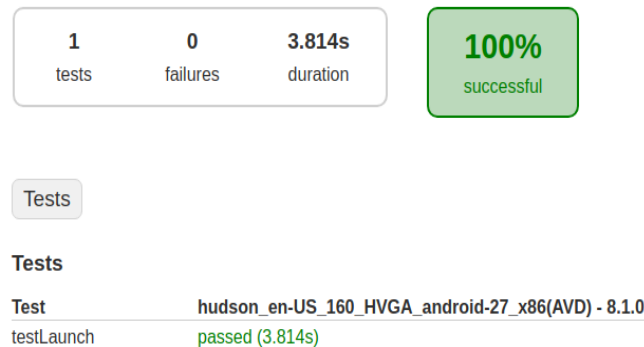
Test Case 2:



**Figure 9 : Jenkins JUnit Tests**

## Class com.example.godam.MainActivityTest

all > com.example.godam > MainActivityTest



Generated by [Gradle 4.10.1](#) at 23 Apr, 2019 12:08:59 PM

*Figure 10 : Jenkins Test Result*

## Continuous Integration and Deployment Tools

**Continuous Integration** (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early. By integrating regularly, we can detect errors quickly, and locate them more easily.

In our project, we have used '**Jenkins**' for continuous integration.

### Jenkins

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

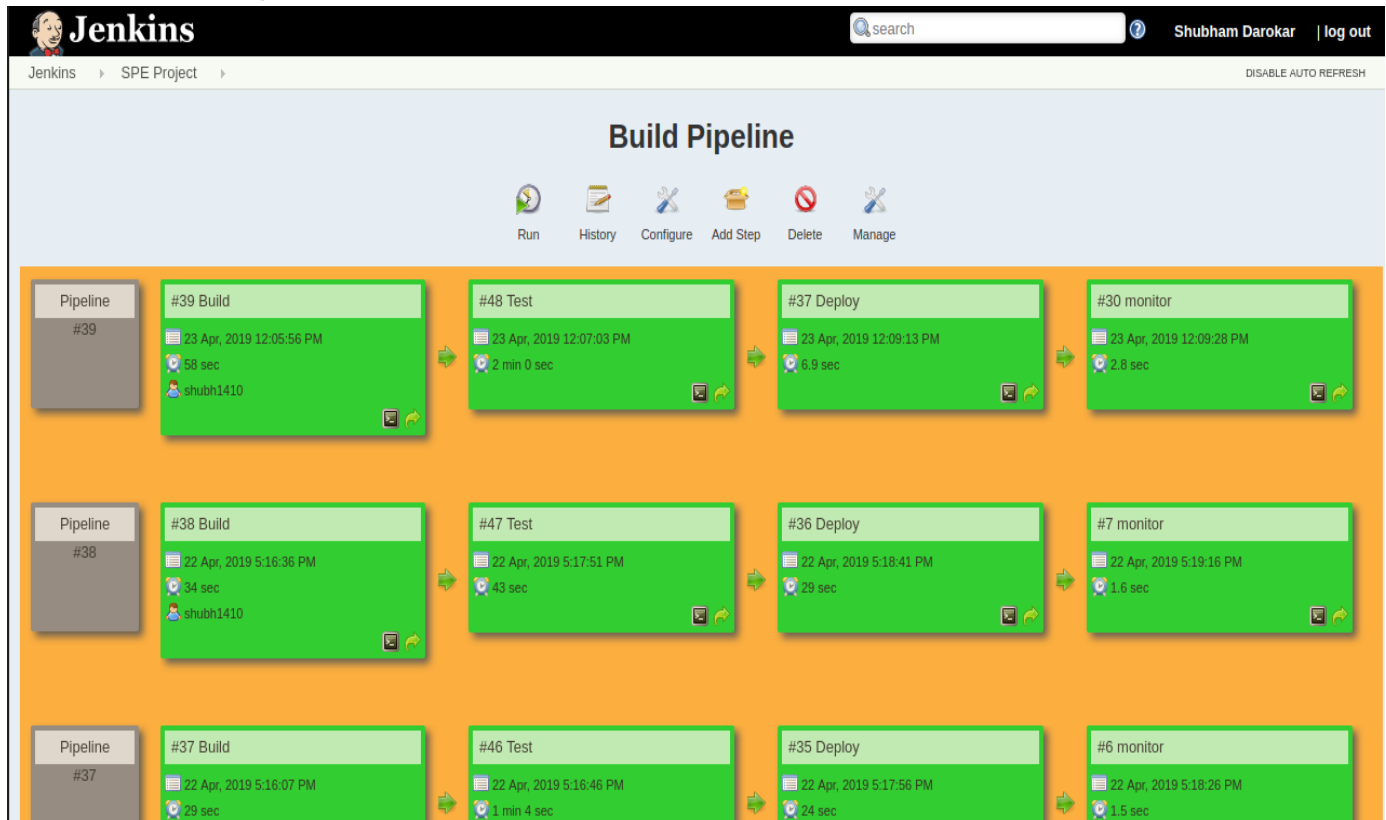
Jenkins is a highly extensible product whose functionality can be extended through the installation of plugins

***/\*To be filled by Shubham\*/***

Different Stages of our pipeline are as follows:



- 1) Build:
- 2) Test:
- 3) Deploy:
- 4) Monitoring:



**Figure 11 : Jenkins Pipeline**

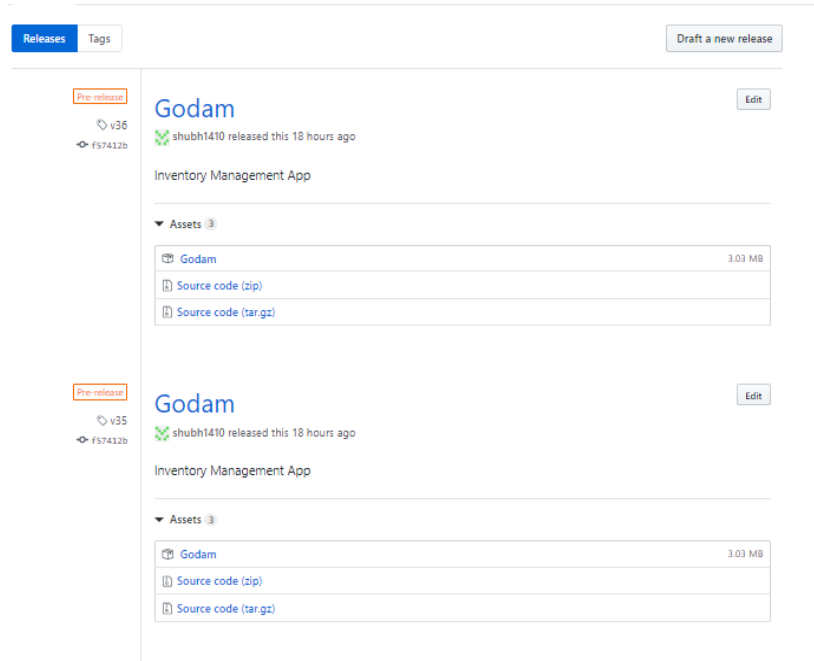
**Continuous Deployment (CD)** is a strategy for software releases wherein any code commit that passes the automated testing phase is automatically released into the production environment, making changes that are visible to the software's users. In our project we are deploying android package file to '**GitHub-Release**'.

### GitHub-Release

We can create a release to package software, along with release notes and links to binary files, for other people to use. Releases are based on Git tags, which mark a specific point in our repository's history. Releases are ordered by the date they are created on GitHub.

**/\*To be filled by Shubham\*/**

**/\*short desc of commands written in deploy node of pipeline \*/**



## Monitoring Tools

Continuous Monitoring is the formal process of defining an agency's IT systems, categorizing each of these systems by the level of risk, application of the controls, continuous monitoring of the applied controls, and the assessment of the effectiveness of these controls against security threats.

In our project, we have used '**ELK Stack**' for continuous monitoring.

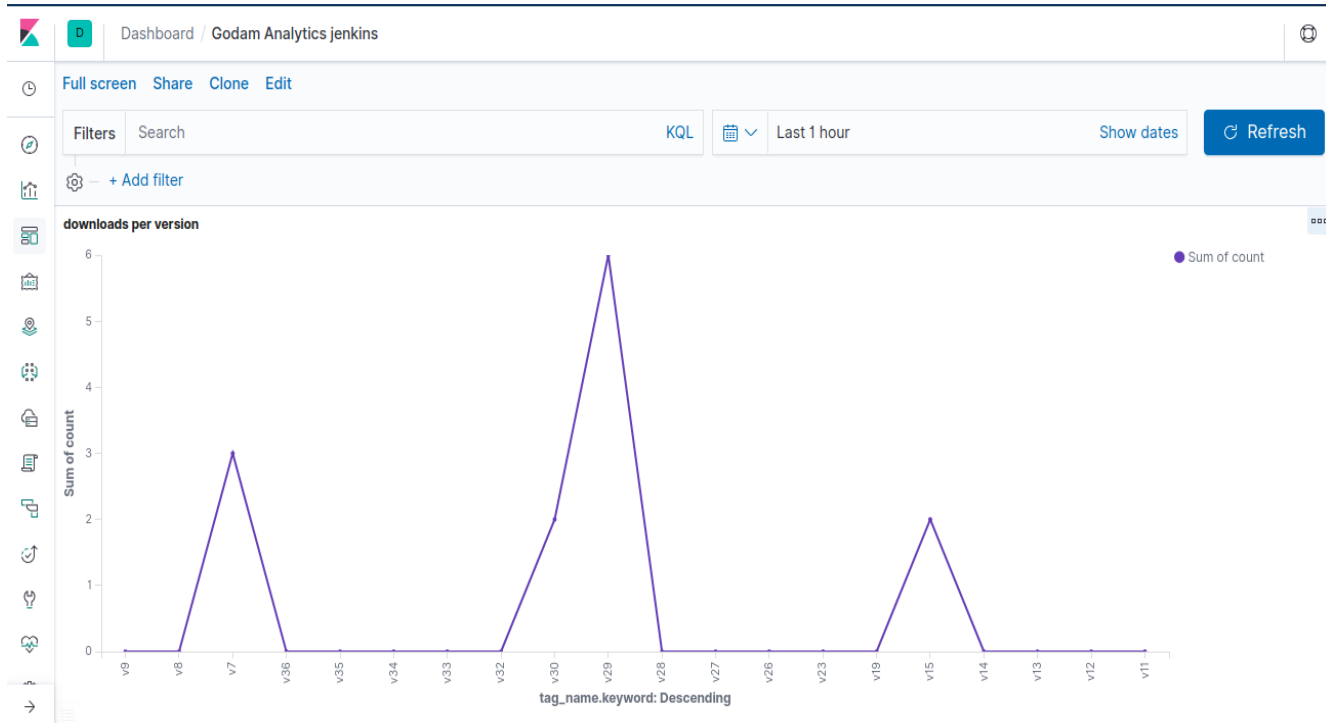
## ELK Stack

"ELK" is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch.

**/\*To be filled by Shubham\*/**

**/\*Steps performed for analytics\*/**

Some of the Kibana Visualization for our project are as follows:



**Figure 12 : Downloads vs Version**



**Figure 13 : Users Created per Day**

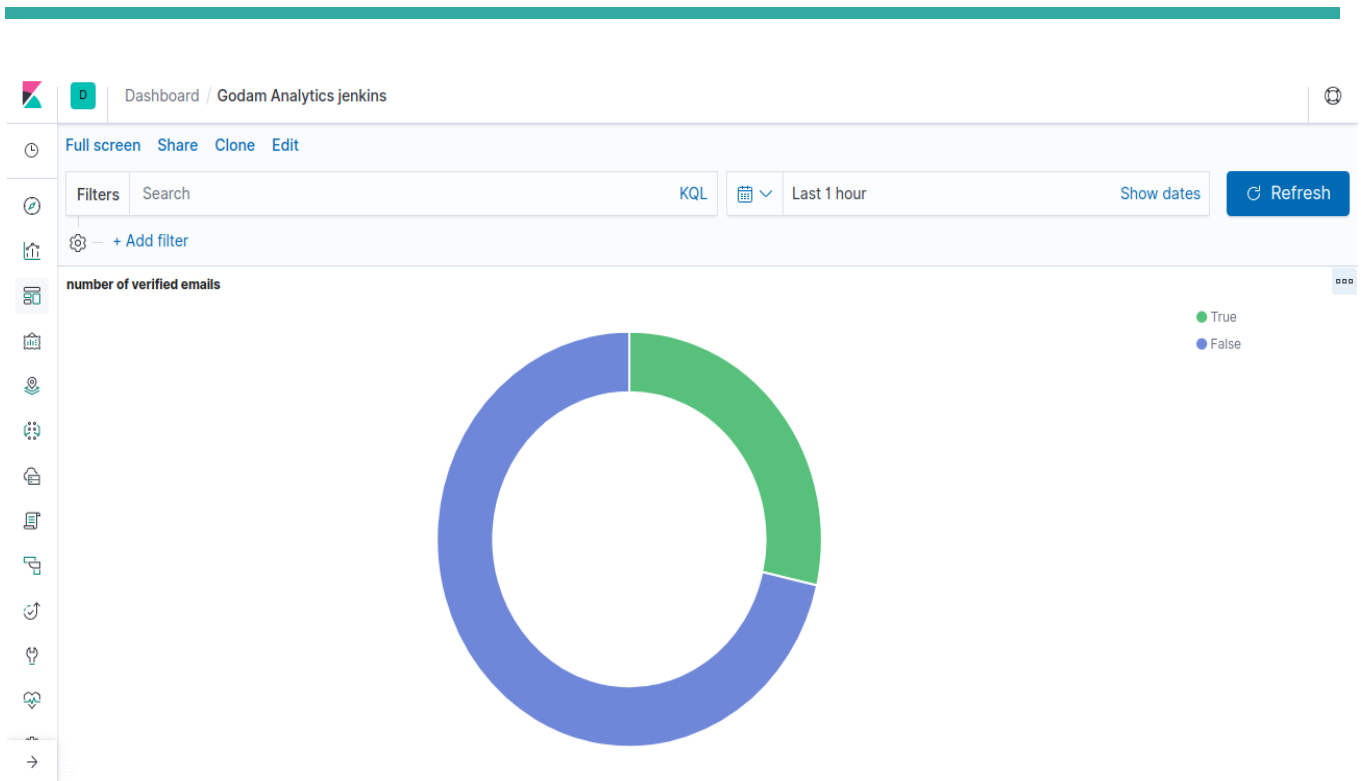


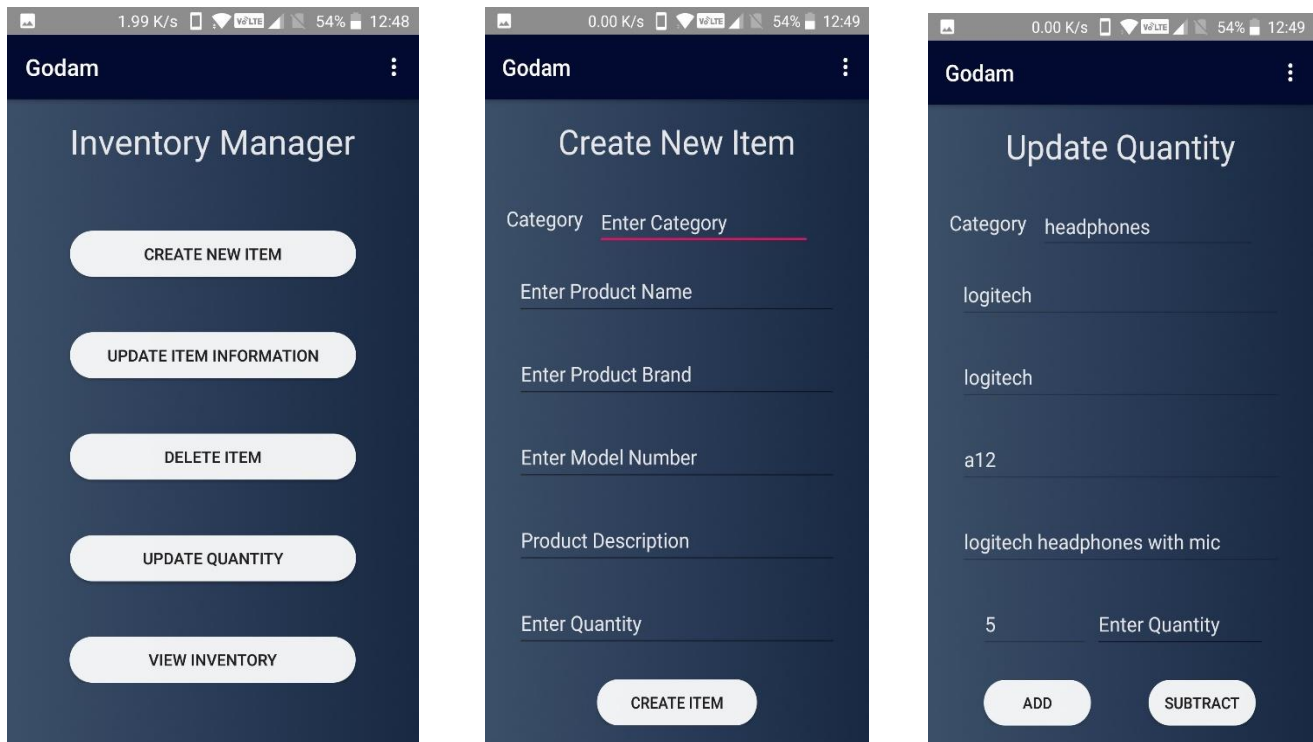
Figure 14 : Number of Verified Emails

## Results

The login screen displays a warehouse icon and input fields for 'User Name' and 'Password'. A 'LOGIN' button is located at the bottom.

The main menu displays two primary actions: 'CREATE NEW USER' and 'VIEW INVENTORY'. A 'Signed in' status indicator is shown at the bottom.

The 'Create New User' screen includes input fields for 'First Name', 'Last Name', 'Phone Number', 'User Name', 'Password', and 'Confirm Password'. A 'Type' dropdown is set to 'Inventory Manager'. A 'CREATE USER' button is at the bottom.



## Future Work

Current version of Godam application bridges the information gap of owner sitting in store and stocks in warehouse but the same application can be extended for use in shops for billing or to know total worth of goods kept in the warehouse. This android application can also be extended to cross check product list supplied by supplier and received at warehouse.

## Conclusion

The application Godam is working flawlessly for businesses to operate warehouses, where owner keeps the records of available stocks. This project eliminates the paperwork, human faults, manual delay and speed up the process. Godam will have the ability to track available stock in real time. This is simple, fast and intelligent stock management that can be used by anyone who has a smartphone.

We have used various open source tools to incorporate DevOps methodology in our application development life cycle namely Git, Jenkins, Gradle, JUnit framework and

---

ELK Stack which made our development cycle smooth and efficient. Errors were detected in initial stages and resolved in the next iterations decreasing the release time of our application product.

---

# List of Figures

Figure 1 : DevOps Culture .....	4
Figure 2 : DevOps Cycle.....	6
Figure 3 : Software Development Life Cycle .....	9
Figure 4 : Use Case Diagram for Supervisor .....	11
Figure 5 : Use Case Diagram for Inventory Manager.....	11
Figure 6 : Godam Git Repository .....	13
Figure 7 : Godam Git Repository Branches .....	13
Figure 8 : Gradle Dependencies.....	14
Figure 9 : Jenkins JUnit Tests.....	15
Figure 10 : Jenkins Test Result .....	16
Figure 11 : Jenkins Pipeline .....	17
Figure 12 : Downloads vs Version .....	19
Figure 13 : Users Created per Day .....	19
Figure 14 : Number of Verified Emails.....	20

---

# References

- 1] SDLC Overview - [https://www.tutorialspoint.com/sdlc/sdlc\\_overview.htm](https://www.tutorialspoint.com/sdlc/sdlc_overview.htm)
- 2] <https://www.hexacta.com/2018/05/07/why-devops-in-two-perspectives/>
- 3] <https://jenkins.io/>
- 4] <https://www.guru99.com/junit-tutorial.html>
- 5] <https://developer.android.com/docs>
- 6] Introduction to Software Production Engineering (2018-T2-CS 816 / Software Production Engineering)
- 7] <https://gradle.org/>
- 8] [https://en.wikibooks.org/wiki/Introduction\\_to\\_Software\\_Engineering/Tools/Build\\_Tools](https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Tools/Build_Tools)
- 9] <https://docs.gradle.org/current/userguide/userguide.html>
- 10] <https://en.wikipedia.org/wiki/Gradle>
- 11] <https://jenkins.io/doc/>
- 12] <https://www.elastic.co/elk-stack>