

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018



## MOBILE APPLICATION DEVELOPMENT LABORATORY

### A MINI PROJECT REPORT

ON

## “INVENTORY MANAGEMENT SYSTEM”

*Submitted in partial fulfillment of the requirement for award of degree*

*of*

### BACHELOR OF ENGINEERING

IN

### COMPUTER SCIENCE AND ENGINEERING

By

**NARAHARI PRASAD B S**

**1EP19CS057**

**NIRANJAN N**

**1EP19CS060**

**NITHIN GOWDA M**

**1EP19CS061**

Under the guidance of

**Dr. Heena Kousar Assoc. Prof. Dept of CSE, EPCET**

**Mrs. Ashwini S. Assoc. Prof. Dept of CSE, EPCET**



**EAST  
POINT COLLEGE OF  
ENGINEERING &  
TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Jnana Prabha, Bidarahalli, Virgo Nagar Post, Bengaluru, Karnataka 560049**

**2021-2022**



**EAST  
POINT COLLEGE OF  
ENGINEERING &  
TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Jnana Prabha, Bidarahalli, Virgo Nagar Post, Bengaluru, Karnataka 560049**

**2021-2022**

## **CERTIFICATE**

This is to certify that the Project report entitled ***INVENTORY MANAGEMENT SYSTEM*** is a bonafide work carried out by **NARAHARI PRASAD B S[1EP19CS057]**, **NIRANJAN N [1EP19CS060]** and **NITHIN GOWDA M [1EP19CS061]** in the partial fulfillment of the requirements of VI semester of **Bachelor of Engineering in Computer Science and Engineering in Visvesvaraya Technological University, Belagavi**, during academic year 2020-2021. It is certified that all corrections/suggestions indicated for the project have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of **MOBILE APPLICATION DEVELOPMENT LAB (18CSMP68)** prescribed for the Bachelor Degree in Engineering.

\_\_\_\_\_  
**Signature of Guide 1**

**Dr. Heena Kousar**

**Assoc. Prof. Dept. of CSE,**

**EPCET, Bangalore**

\_\_\_\_\_  
**Signature of Guide 2**

**Mrs. Ashwini S.**

**Assoc. Prof. Dept of CSE,**

**EPCET, Bangalore**

\_\_\_\_\_  
**Signature of HOD**

**Dr. C Emilin Shyni**

**Head of the Dept. CSE,**

**EPCET, Bangalore**

**Name of the Examiner**

**1 .....**

**2 .....**

**Signature with date**

**.....**

**.....**

## **ACKNOWLEDGENT**

Firstly, I thank the Management and Principal of East Point College of Engineering and Technology, Bangalore for providing me an opportunity to work on this project as part of My research work. It gives us immense pleasure to express our deep sense of gratitude whose words of advice have always been a constant source of inspiration for us.

I would like to express my heartfelt thanks to Dr. C Emilin Shyni, Professor and Head of Department of Computer Science and Engineering, EPCET for her valuable advice and encouragement to us in completing this project work.

I am obliged to Mrs. Heena Kousar, Assistant Professor, Dept. of CSE, who rendered valuable assistance as the project coordinators.

I would like to thank my Parents and Friends for their support, encouragement during the course of my project. Finally, I offer my regards to all the faculty members of CSE department and all those who supported me in any respect during the project.

**NARAHARI PRASAD B S [1EP19CS057]**

**NIRANJAN N [1EP19CS060]**

**NITHIN GOWDA M [1EP19CS099]**

## **ABSTRACT**

This project is aimed at developing a desktop based application named Inventory Management System for managing the inventory system of any organization. The Inventory Management System (IMS) refers to the system and processes to manage the stock of organization with the involvement of Technology system. This system can be used to store the details of the inventory, stock maintenance, update the inventory based on the sales details, generate sales and inventory report daily or weekly based. This project is categorize individual aspects for the sales and inventory management system. In this system we are solving different problem affecting to direct sales management and purchase management. Inventory Management System is important to ensure quality control in businesses that handle transactions resolving around consumer goods. Without proper inventory control, a large retail store may run out of stock on an important item. A good inventory management system will alert the wholesaler when it is time to record. Inventory Management System is also on important means of automatically tracking large shipment. An automated Inventory Management System helps to minimize the errors while recording the stock.

# INDEX

CHAPTER NUMBER	CHAPTER NAME	PAGE NO
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
3	REQUIREMENT SPECIFICATION	3
4	ANALYSIS	4
5	SYSTEM DESIGN	5
6	IMPLEMENTATION	6-17
7	TESTING	18
8	SNAPSHOTS	19-20
9	PERFORMANCE ANALYSIS	21-22
10	CONCLUSION	23
11	BIBLIOGRAPHY	24

# LIST OF FIGURES

FIG NO	DESCRIPTION	PAGE NO
FIG 5.1	IMS PROCESS FLOW DIAGRAM	5
FIG 5.2	IMS USE CASE DIAGRAM	5
FIG 8.1	REGISTER PAGE	19
FIG 8.2	LOGIN PAGE	19
FIG 8.3	WELCOME PAGE	19
FIG 8.4	ADD PRODUCT	20
FIG 8.5	SEARCH PAGE	20
FIG 8.6	VIEW INVENTORY	20
FIG 8.7	DELETE ITEM	20

# CHAPTER-1

## INTRODUCTION

Inventory control is very important to an organization because all the stock must be managed, controlled and tracked at anytime and anywhere. Nowadays, inventory control is even more crucial to success and sustainability. Inaccurate inventory control can lead to lost business profits. Chuang and Oliva [1] claimed that IRI (Inventory record inaccuracy) reduces a company's total profits through a study on a retail store which has an IRI of 29% causing the company losing 10% profits. Manually managing the stock may be inaccurate, consume time and also prone to human error. Among the factors causing the inventory record inaccuracy; (i) missing main sources of stock (ii) misplacement of stock and (iii) stock transaction errors [2] To solve these issues, inventory control requires a computerized system to better manage the inventory [3]. With the increased amount of material variations and quantity, manually tracking and counting inventory is nearly impossible. Hence, there are many types of inventory computer software available in the market. However, relying on inventory control and tracking methods solely on a desktop computer is no longer feasible. In 2017, there are almost 20 million smart phone uses in Malaysia [4]. Thus, there is a vast opportunity to develop application software using smart phones for user convenience. There are 101 laboratories in the Faculty of Engineering Technology, University Technical Malaysia Melaka. These laboratories require various consumables materials for the laboratory activities. The record keeping and retrieving were done manually using a yellow form and an Excel spreadsheet. The location of the various stores also added to the issue of sharing information between the departments and tracking similar inventories. These consumable goods need proper management in order to ensure that the students' learning process runs smoothly at all times but still keeping the costs of keeping the inventory at a minimum. Thus, the aim of this study is to develop a system to accurately manage these consumable goods in storage. The use of the E-Inventory system has helped to improve the efficiency of the laboratory consumables materials management.

## CHAPTER-2

### LITERATURE SURVEY

Products are considered as the business resources for the organization. This includes managing the product with appropriate way to review any time as per the requirement. Therefore it is important to have a computer based IMS which has the ability to generate reports, maintain the balance of the stock, details about the purchase and sales in the organization. Before developing this application we came up with 2 Inventory Management System existing in the market, which helps to give the knowledge for the development of our project. These application software are only used by the large organization but so we came up with the application which can be used by the small company for the management of their stock in the production houses. After analyzing the other inventory management system we decided to include some of common and key features that should be included in every inventory management system. So we decided to include those things that help the small organization in a way or other.

### 2.1 Advantages of proposed system:

After analyzing many existing IMS we have now the obvious vision of the project to be developed. Before we started to build the application team had many challenges.

We defined our problem statement as:

- To make desktop based application of IMS for small organization.
- To make the system easily managed and can be secured.
- To cover all the areas of IMS like purchase details, sales details and stock management.

### 2.2 Scope of the Application:

Inventory Management System (IMS) is targeted to the small or medium organization which doesn't have many godwom or warehouses i.e. only to those organization that has single power of authority. Some of the scope are:

- Only one person is responsible in assigning the details or records .
- It is security driven.
- Godown can be added as per the requirement.

## **CHAPTER 3**

### **SYSTEM REQUIREMENT AND SPECIFICATION**

The hardware and software requirements are very minimal and the software can run on most of the machine even of the past . Here we have used the system of below specification to develop.

#### **3.1 HARDWARE REQUIREMENTS**

- PROCESSOR :Intel CORE i5
- RAM : 4GB
- HARD DISK SPACE : 1TB
- MOUSE AND KEYBOARD

#### **3.2 SOFTWARE REQUIREMENTS**

- JDK 1.8 or above.
- Android Studio SDK.
- Language: JAVA, XML



## CHAPTER 4

### SYSTEM ANALYSIS

#### 4.1 Background Research :

We started research by identifying the need of IMS in the organization. Initially we bounded our research to find the general reasons that emerged the needs of Inventory Management System. We used different techniques to collect the data that can clearly give us the overall image of the application. The techniques we used were interview with the developers, visiting online websites that are presented as the templates and visiting some organization to see their IMS application. Basically the following factors forced us to develop IMS application:

- Cost and affordability
- Lack of stock management.
- Effective flow of stock transfer and management.
- Difficulty in monitoring the stock management.

#### 4.2 Requirement Analysis:

We collected a number of requirements for project from our primitive research, website visits, and interview to the concerned personnel and their experiences regarding the concepts of its development. We have even visited some organization in Kathmandu valley and analyze its importance and try to develop the project by fulfilling all the weakness that were found in the application. We then decided to build same type of application with different logic flow and new language which will be suitable for the small organization.

#### 4.3 IMS Requirement:

The goal for the application is to manage the inventory management function of the organization. Once it is automated all the functions can be effectively managed and the organization can achieve the competitive advantage. Business requirement are discussed in the Scope section, with the following additional details:

- Helps to search the specific product and remaining stock.
- Details information about the product sales and purchase.
- It helps to identify the total presented inventory in the company.

## CHAPTER 5

### SYSTEM DESIGN

Process Flow Diagram Process Flow Diagram or Flowchart is a diagram which uses geometric symbols and arrows to define the relationships. It is a diagrammatic representation of the algorithm. The Process flow Diagram of our application is shown below:

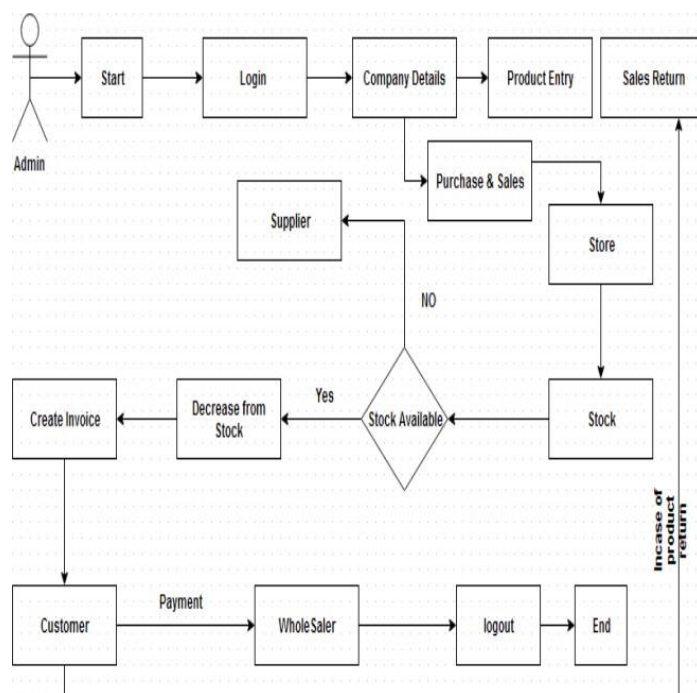


FIG 5.1 IMS Process flow diagram

### 5.1 Use Case Diagram

Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors and their goals. The main purpose of a use case diagram is to show what system functions are performed for which actors.

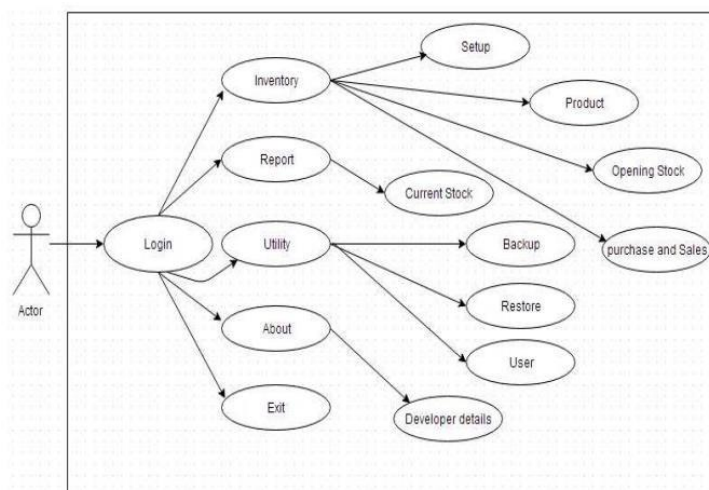


FIG5.2 IMS Use Case Diagram

# CHAPTER 6

## IMPLEMENTATION

App's user interface is everything that the user can see and interact with. Android provides a variety of pre-built UI components such as structured layout objects and UI controls that allow us to build the graphical user interface for the app.

The UI design can be done in 2 ways.

- One is using plug and play where in we directly design using the inbuilt controls such as EditText, TextView, Buttons etc..
- Second method is by directly writing XML code.

### XML CODE:

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.example.admin.augscan">

    <uses-permission android:name="android.permission.CAMERA" />

    <uses-feature android:name="android.hardware.camera" />

    <application

        android:allowBackup="true"

        android:icon="@mipmap/ic_launcher"

        android:label="WareHouse App"

        android:roundIcon="@mipmap/ic_launcher_round"

        android:supportsRtl="true"

        android:theme="@style/AppTheme">

        <activity android:name=".ScanCodeActivitysearch"></activity>

        <activity android:name=".ScanCodeActivity" />

        <activity android:name=".ScanCodeActivitydel" />

        <activity android:name=".viewWareHouseActivity" />
```

```

<activity android:name=".scanItemsActivity" />

<activity android:name=".deleteItemsActivity" />

<activity android:name=".additemActivity" />

<activity android:name=".dashboardActivity" />

<activity android:name=".RegisterActivity" />

<activity android:name=".LoginActivity" />

<activity android:name=".MainActivity" />

<activity

    android:name=".SplashsActivity"

    android:theme="@style/AppTheme.AppCompat.NoActionBar">

    <intent-filter>

        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />

    </intent-filter>

</activity>

<meta-data

    android:name="preloaded_fonts"

    android:resource="@array/preloaded_fonts" />

</application>

</manifest>

```

## JAVA CODE:

```

package com.example.admin.augscan;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

```

```

import android.view.View;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

public class MainActivity extends AppCompatActivity {

    private FirebaseAuth auth;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        auth = FirebaseAuth.getInstance();

        FirebaseUser user = auth.getCurrentUser();

        if(user != null){

            finish();

            startActivity(new Intent(this, dashboardActivity.class));

        }

    }

    public void login (View view)

    {

        startActivity(new Intent(this,LoginActivity.class));

    }

    public void register (View view)

    {

        startActivity(new Intent(this,RegisterActivity.class));

    }

}

```

```
//Registre
```

```
package com.example.admin.augscan;
```

```
import android.content.Intent;
```

```
import android.support.annotation.NonNull;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
import android.text.TextUtils;
```

```
import android.util.Log;
```

```
import android.util.Patterns;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
import android.widget.ProgressBar;
```

```
import android.widget.Toast;
```

```
import com.google.android.gms.tasks.OnCompleteListener;
```

```
import com.google.android.gms.tasks.Task;
```

```
import com.google.firebase.auth.AuthResult;
```

```
import com.google.firebase.auth.FirebaseAuth;
```

```
import com.google.firebase.auth.FirebaseUser;
```

```
import com.google.firebase.auth.UserProfileChangeRequest;
```

```
import com.google.firebase.database.FirebaseDatabase;
```

```
public class RegisterActivity extends AppCompatActivity {
```

```
    private EditText editTextName, editTextEmail, editTextPassword, editTextPhone, editTextcPassword;
```

```
    public Button UserRegisterBtn;
```

```
    private ProgressBar progressBar;
```

```

private FirebaseAuth mAuth;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_register);

    editTextName = findViewById(R.id.departmentName);

    editTextEmail = findViewById(R.id.emailRegister);

    editTextPassword = findViewById(R.id.passwordRegister);

    editTextcPassword= findViewById(R.id.confirmPassword);

    UserRegisterBtn= findViewById(R.id.button_register);

    progressBar = findViewById(R.id.progressbar);

    progressBar.setVisibility(View.GONE);

    mAuth = FirebaseAuth.getInstance();

    UserRegisterBtn.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            registerUser();

        }

    });

}

@Override

protected void onStart() {

    super.onStart();

    if (mAuth.getCurrentUser() != null) {

        }
    }

```

```

}

private void registerUser() {

    final String name = editTextName.getText().toString().trim();

    final String email = editTextEmail.getText().toString();

    String password = editTextPassword.getText().toString().trim();

    String cpassword = editTextcPassword.getText().toString().trim();

    if (email.isEmpty()) {

        editTextEmail.setError("It's empty");

        editTextEmail.requestFocus();

        return;

    }

    if (name.isEmpty()) {

        editTextName.setError("It's Empty");

        editTextName.requestFocus();

        return;

    }

    if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {

        editTextEmail.setError("Not a valid emailaddress");

        editTextEmail.requestFocus();

        return;

    }

    if (password.isEmpty()) {

        editTextPassword.setError("Its empty");

        editTextPassword.requestFocus();

        return;

```



```

}

if (password.length() < 6) {

    editTextPassword.setError("Less length");

    editTextPassword.requestFocus();

    return;

}

if(!password.equals(cpassword)){

    editTextcPassword.setError("Password Donot Match");

    editTextcPassword.requestFocus();

    return;

}

progressBar.setVisibility(View.VISIBLE);

mAuth.createUserWithEmailAndPassword(email, password)

    .addOnCompleteListener(new OnCompleteListener<AuthResult>() {

        @Override

        public void onComplete(@NonNull Task<AuthResult> task) {

            if (task.isSuccessful()) {

                final User user = new User(

                    name,

                    email

                );

                FirebaseUser usernameinfirebase = mAuth.getCurrentUser();

                String UserID=usernameinfirebase.getEmail();

                String resultemail = UserID.replace(".", "");

                FirebaseDatabase.getInstance().getReference("Users")

```

```

        .child(resultemail).child("UserDetails")

        .setValue(user).addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

    progressBar.setVisibility(View.GONE);

    if (task.isSuccessful()) {

Toast.makeText(RegisterActivity.this, "Registration Success", Toast.LENGTH_LONG).show();

        startActivity(new Intent(RegisterActivity.this, dashboardActivity.class));

    } else {

        //display a failure message

    }

}

});

} else {

    progressBar.setVisibility(View.GONE);

    Toast.makeText(RegisterActivity.this, "Registration Failed",
Toast.LENGTH_LONG).show();

    }

}

});

}

}

// Login

package com.example.admin.augscan;

import android.app.ProgressDialog;

```

```
import android.content.Intent;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

public class LoginActivity extends AppCompatActivity {

    private EditText Email;

    private EditText Password;

    private Button Login;

    private TextView passwordreset;

    private EditText passwordresetemail;

    private ProgressBar progressBar;

    private FirebaseAuth auth;

    private ProgressDialog progressDialog;

    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_login);

    Email = (EditText) findViewById(R.id.emailSignIn);

    Password = (EditText) findViewById(R.id.password);

    Login = (Button) findViewById(R.id.Login);

    passwordreset = findViewById(R.id.forgotpassword);

    passwordresetemail = findViewById(R.id.emailSignIn);

    progressBar = (ProgressBar) findViewById(R.id.progressbars);

    progressBar.setVisibility(View.GONE);

    auth = FirebaseAuth.getInstance();

    progressDialog = new ProgressDialog(this);

    Login.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View view) {

            validate(Email.getText().toString(), Password.getText().toString());

        }

    });

    passwordreset.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            resetpassword();

        }

    });

}

```

```

public void resetpassword(){

    final String resetemail = passwordresetemail.getText().toString();

    if (resetemail.isEmpty()) {

        passwordresetemail.setError("It's empty");

        passwordresetemail.requestFocus();

        return;

    }

    progressBar.setVisibility(View.VISIBLE);

    auth.sendPasswordResetEmail(resetemail)

        .addOnCompleteListener(new OnCompleteListener<Void>() {

            @Override

            public void onComplete(@NonNull Task<Void> task) {

                if (task.isSuccessful()) {

                    Toast.makeText(LoginActivity.this, "We have sent you instructions to reset your
password!", Toast.LENGTH_SHORT).show();

                } else {

                    Toast.makeText(LoginActivity.this, "Failed to send reset email!",
Toast.LENGTH_SHORT).show();

                }

                progressBar.setVisibility(View.GONE);

            }

        });

    }

    public void validate(String userEmail, String userPassword){

        progressDialog.setMessage(".....Please Wait..... ");

        progressDialog.show();

```

```
auth.signInWithEmailAndPassword(userEmail, userPassword).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {

    @Override

    public void onComplete(@NonNull Task<AuthResult> task) {

        if(task.isSuccessful()){

            progressDialog.dismiss();

            Toast.makeText(LoginActivity.this, "Login Successful", Toast.LENGTH_SHORT).show();

            startActivity(new Intent(LoginActivity.this, dashboardActivity.class));

        }

        else{

            Toast.makeText(LoginActivity.this,"Login Failed", Toast.LENGTH_SHORT).show();

            progressDialog.dismiss();

        }

    }

});

}
```

# CHAPTER 7

## TESTING

The purpose of software testing is to access or evaluate the capabilities or attributes of a software program's ability to adequately meet the applicable standards and application need. Testing does not ensure quality and the purpose of testing is not to find bugs. Testing can be verification and validation or reliability estimation. The primary objective of testing includes: —

- To identifying defects in the application.
- The most important role of testing is simply to provide information.
- to check the proper working of the application while inserting updating and deleting the entry of the products.

### 7.1Type of Testing

We have used one type of testing to ensure the error free features of our software application:

#### 7.2.1 Units Test

This type of testing is the testing of individual software components. It is typically done by the programmer and not by the testers. It requires details information and knowledge about the internal program design and code to perform this. During unit testing, we carried out various testing task such as the reflection of the unit data on database and its interface. Various types of bugs associated with the component were identified and fixed. We use various functional keys to test our software.

In our software unit testing is concerned with the stock units, opening stock units and product units validation as well as the validation of product units.

# CHAPTER 8

## SNAPSHOTS

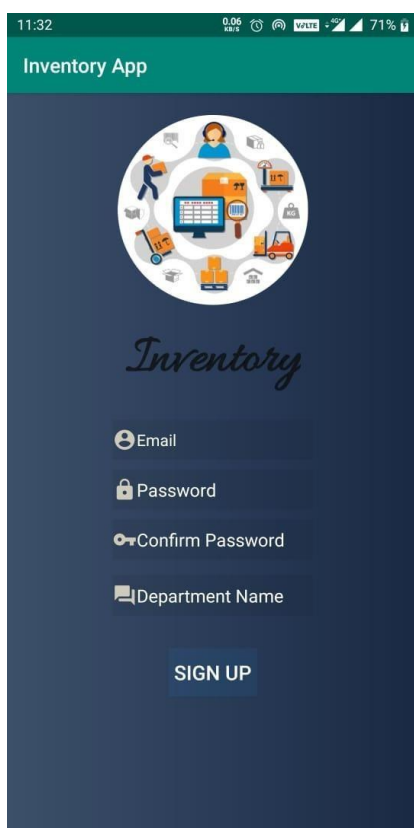


FIG8.1:REGISTER PAGE

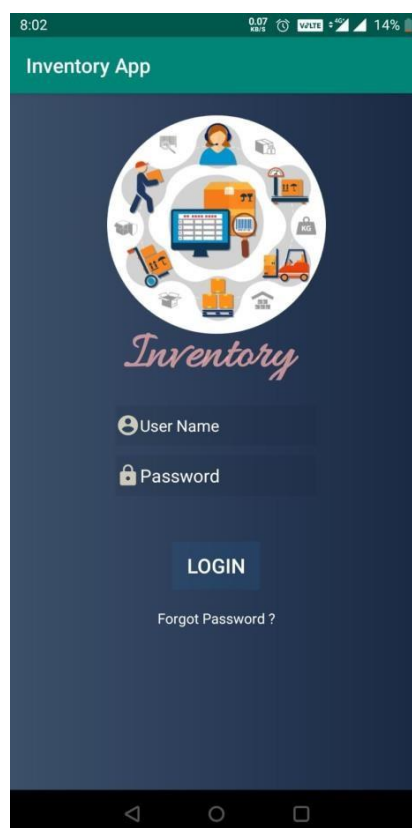


FIG8.2 :LOGIN PAGE

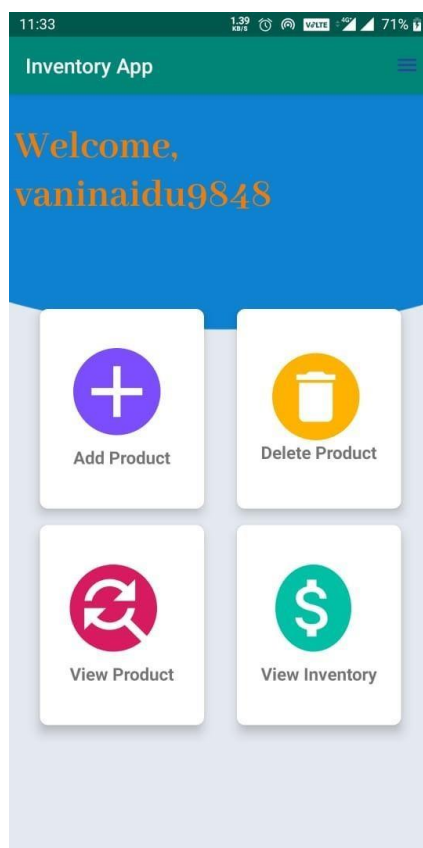


FIG 8.3:WELCOME PAGE



Inventory App

# Product Details

Product

Category

Price

Number

FIG 8.4:ADD PRODUCT

Inventory App

# Scan Items

Item Barcode	8906059320101
Item Name	HP
Item Price \$	30
Item Category	Laptop

FIG 8.5: SEARCH ITEM

Inventory App

No of Items = 1    Total sum = \$ 30

Item Barcode	8906059320101
Item Name	HP
Item Price \$	30
Item Category	Laptop

FIG 8.6:VIEW INVENTORY

Inventory App

# Delete Items

Item to Delete:

FIG 8.7: DELETE ITEM

## CHAPTER 9

### PERFORMANCE ANALYSIS

Software performance analysis looks at how a specific program is performing on a daily basis and chronicles what slows down performance and causes errors now and what could pose a problem into the future. Performance issues aren't always built into software in a way that can easily be spotted through the QA process. Instead, it is something that can emerge over time after the project has been deployed. Software performance analysis keeps your team honest: it requires developers to continually test what they are doing and IT teams to monitor the code as version are updated, more code is added, other applications interact with it, or when there are changes in hosting. While many businesses will perform software performance analysis at random times, that isn't enough. Performance is something that needs to be constantly monitored for problems because they aren't always apparent and catching something early (perhaps even before the end user is impacted) can save a lot of time and strife.

- **Software Performance Evaluations & Analysis Eliminate Rework** When designing new applications or even making changes to existing ones, there are bound to be some errors or things that go wrong. This can impact software performance immediately or it can impact it as a slow leak over time. Either way, it should be caught as soon as possible. While those immediate problems may seem like the most hazardous, they don't have to be. The slower, smaller problems that compound over time can cause eventual crashes or a lapse in security that can bring down your entire portfolio. When you think of all that could be stopped by a simple mistake - web applications, mobile applications, internal operations, and so much else, it can be disconcerting. The software performance needs to meet the performance requirements of today's world, regardless of when they were created. Sometimes, trying to keep up can fall by the wayside and reveal many, many problems. When you let something go too long, the answer may be losing all of that work and trying to develop something completely new: losing everything that you have done already. To avoid this, software performance analysis needs to be automated into every stage of the process. The IT teams and developers need to get their work right the first time and eliminate the need for a lot of rework. By adding software performance analysis into your existing culture, you will see changes that result in more productivity and even more employee satisfaction.
- **Software Performance Analysis Helps Businesses Achieve Optimization** Of course, IT teams are expected to perform at a certain level and take a certain amount of pride in their work. Still, we all make mistakes and when work becomes rote or repetitive as coding can be, mistakes are bound to happen. Everyone must take steps to ensure that they are working toward goals with efficiency: the work needs to be good and it needs to get done quickly. With software performance analysis, they will have some back up in checking their work. It is important to note that this is far more than

simplistic testing. Instead, it is complete, comprehensive monitoring of code throughout the lifespan of the application.

- **Incorporate Software Performance Analysis into Your System** At the end of the day, software performance analysis will help to simulate how a system will perform today, tomorrow, three weeks from now, and into the next year. It checks for potential problems that may emerge when you test it against traffic, load conditions, and business requirements. The goal is to reduce technical debt while at the same time providing a better business value.

## CHAPTER 10

### CONCLUSION

To conclude, Inventory Management System is a simple desktop based application basically suitable for small organization. It has every basic items which are used for the small organization. Our team is successful in making the application where we can update, insert and delete the item as per the requirement. The E-Inventory system was successfully developed and tested to perform the intended function for the manufacturing department's fast and accurate inventory management system. Further improvement includes added new feature to access camera to take and store the material photo and to add chat feature to the system for user to leave a message to the store keeper to request material. Further, functions to import and export data for report generation should also be included. Since currently only the Android version is available, future improvement should also allow E-Inventory to be accessed using the iOS application. This application also provides a simple report on daily basis to know the daily sales and purchase details. This application matches for small organization where there small limited if godwoms. Through it has some limitations, our team strongly believes that the implementation of this system will surely benefit the organization.

## **BIBLIOGRAPHY**

### **Software Reference**

- Swatik Accounting And Inventory Software
- High-tech Software, Kalimati
- Inventory Management Software Sagar International, Balkhu

### **Website**

- Visual Studio Official Site: <https://msdn.microsoft.com/en-us/library/dd492171.aspx>
- <https://www.w3schools.com>
- <https://stackoverflow.com>