Warehouse Management System

In Ruby On Rails 5

Akash Sadanand Hande
MSc. in Cloud Computing
National College of Ireland
X17156220@student.ncirl.ie

Pulkit Garg
MSc. in Cloud Computing
National College of Ireland
X17156882@student.ncirl.ie

***Abstract*— This report describes a web-based application for managing multiple warehouses developed in Ruby on Rails 5 framework and delivered as software as a Service (SaaS). The application follows the MVC architecture with SQLite as a database on development and Postgre at the production stage. The application is deployed on Heroku cloud application platform. This application delivers responsive webpages which can be accessed on any type of computing devices from laptop to mobile phones as per of their screen size. This application helps the customer to manages multiple warehouses each with their independent functioning and has separate products, sources, and categories which are approved by the admin and can perform order-in, order-out operations, generate reports such as inventory on hand, shipments by the customer, product shortage. Each user of WMS can be associated with a single warehouse and can't access the data of another warehouse. The admin has access to the dashboard and can access information about all the warehouses and can add new products, sources, categories to the system for each warehouse. The admin can access advance dynamic information such as location wise sales data.**

I. INTRODUCTION

The warehouse is a commercial building for efficient storage and handling of goods and materials. The warehouse can be used by a different entity such as e-commerce companies, distributors, manufacturers, logistic suppliers and is a key element to efficiently run any business which needs to store physical goods.

In today's digital era, the warehouse is not just about storage and retrieval of goods. Modern businesses need to have multiple warehouses strategically located across the market to process the goods efficiently while minimizing the cost maximizing the space utilization. Warehouse management system provides key information about an organization product inventory and location at any movement.

Application login details & Heroku Deployment Links mentioned below:

Akash Hande - https://wms-akash.herokuapp.com/

Pulkit Garg - https://wms-pulkit.herokuapp.com/

User ID	*Password*	*Role*
admin@wms.com	Adminware0	Admin
Wh1@wms.com	Warehouse1	The user of Warehouse 1
Wh2@wms.com	Warehouse2	The user of Warehouse 2
Wh3@wms.com	Warehouse3	The user of Warehouse 3

Table 1.1 Sign in credentials

A. Dashboard

The Dashboard is dynamic in nature and display information based on the logged in users it displays data from all the warehouse such as a total number of products, sources, and categories. there is a drop-down menu which generates a prediction for each product available in database and statistics of three warehouses implemented in the system. For the end user the dashboard display data for the logged in users such as warehouse total incoming - outgoing orders and upcoming deliveries for tomorrow.

B. Navigation Bar

The application has a dynamic navigation bar on the top and the left side which are available on all the pages. on the right, there is a link to the login page and after a user is logged in the sidebar display links to the pages which are authorized for the user.

C. Order Ins

This page displays all the records of incoming orders in the warehouse and the options to edit or delete any record. , There is a button to create a new order and after creating the product quantity, selling and purchase price will be updated to the database and is updated to the database and a receipt is generated for the user.

D. Order Out

the interface of this page is like that of order ins page while generating a new outgoing order additional delivery information is also required. after success full creation of the order, the product quantity is reduced from the database.

E. Reports

This page is available for both the admin and the user for the admin the user can select the type of report required by using the drop-down menu and download it by using the generate report button.

F. Sources

This page is used to enter a new source for the product for a warehouse the admin can view all the existing products in the database and assign them to a warehouse.

G. Categories

The page is added to the application to classify the product into the categories for better management and storage requirement.

H. Product

The product page displays the current inventory of the warehouse. The admin can create a new product for a warehouse and can update the quantity and prices of the goods.

The below table illustrates the role-based authorization for the pages to the application:

Module	Admin	User
Dashboard	Yes	Yes
Order ins	No	Yes
Order outs	No	Yes
Products	Yes	No
Sources	Yes	No
Categories	Yes	No
Reports	Yes	Yes
User Profile	Yes	No

Table 1.2 Role-based access to modules in WMS

II. Application Goals

The developed software allows users to manage the day to day operations of the warehouse and to perform analysis of the current inventory scattered in a different warehouse. WMS being a web application makes it possible to access it with any device capable of running any one of the popular web browsers. as the web pages are developed to be responsive in nature it is possible to access the web pages on any screen size the application will automatically scale it to fit the screen. When a user accesses the services on the website, the user will have to sign in with two credentials - user email id and password. Each user upon registering has a warehouse/company assigned to it and only the admin has access to the user sign-up page where it is mandatory to assign a company to the user. The WMS application is distributed as SaaS and can have as many companies and used as required by the organization. Distributing application in SaaS has multiple advantages it offers easy software distribution for the developers and reduces cost. being on a cloud all the code and database are kept in a single place and it is very convenient to upgrade the system which can be reflected all of the end users.

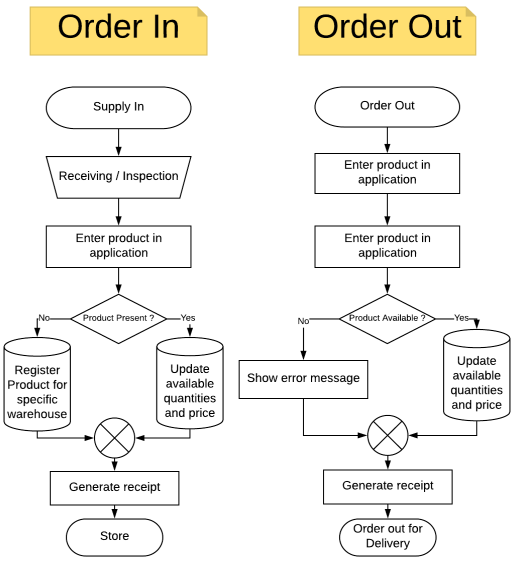

Figure 2.1 Flow chart of daily operations [16]

A. Functional Features:

The WMS software has multiple features for the end user it can generate receipts after creating and updating an order. The system will ask for confirmation whenever the end user wants to delete an order to avoid human mistakes. The system has a parent entity called admin which has the option to add, modify or delete the products, sources, and category for each of the following warehouses. As this software is meant only for commercial use it is mandatory that the new user creation and warehouse assignment can only be authorized by the admin superuser to limit the unauthorized use of the system. The system will keep a record of each entry of the system and associate it with the warehouse. A user of a warehouse does not have access to the data of another user.

B. Business features

WMS System has a multi-company model under which each company has some specific products that come from a unique source and categories which are assigned by the admin. that are only available to the specific end user of WMS software. For the admin, there is a dashboard which represents the following data such as total incoming and outgoing orders for all the warehouses. notify the users about the deliveries which are pending for tomorrow. WMS provides a prediction for sale of products based on existing records according to the geographical region.

C. Reporting Feature

Reports are an important part of any ERP enterprise resource planning system, traditionally reports are created manually by a user with the help of tools like spreadsheet and then shared with the concerned department, but WMS application has a dynamic feature which generates reports for

the individual warehouse or for the admin in Pdf format for the convenience of printing and emailing. WMS offers user three different types of reports mentioned below:

1) Inventory in Hand - WMS report on the amount of inventory available for shipment at any given time. This report can be used to monitor the stock levels or for auditing purpose to prevent miscounts, theft, or other mishaps.

2) Daily Incoming Report - WMS report for documenting the daily incoming operations of the warehouse this report is useful when evaluating the order processing capability of the warehouses.

3) Products Shortage - Report for notifying the user about the products which are running low in stock this report is useful for notifying the clients for reordering purposes.

III. DESIGN

Software application architecture is the process of defining the high-level structure of software solution which endeavors to fill the gap between business requirement and technical requirement while improving attributes such as security, maintainability, and performance [1]. The goal for software architecture is to understand the high-level business requirement, design behavior of application before actual development of the application, use cases, the interaction between various entities in the application, reuse of component in the application and most importantly it affects system development, deployment, and maintenance. Here is the brief description of WMS architectural design:

A. System Architecture and Framework Adoption

A software framework is a platform for developing software applications which provide a foundation to software developers for building their program for specific platform [2]. The main objective of the framework is to provide structure to the application, provides core functionality to the application such as CRUD, session management etc. which saves time and effort and provides common operations in the application. In today's technical era, developers have large variety choices of programming languages to develop their application with their unique features such as object-oriented behavior, scalability, portability, exception handling etc. From the developer's point of view, we chose Ruby as a programming language with "Ruby on Rails" framework for the development of "Warehouse Management System" for its decent features as listed below:

- Model View Controller architecture.
- Single command scaffolding and easy DB migration.
- Object Relational Mapping.
- Database access library.
- With template system, you can easily integrate presentation layer technologies such as HTML, CSS, jQuery, AJAX etc.
- Easily integration of external libraries for a common task.
- Easier for debugging.
- Modularize development of component.
- Good security and performance.

B. System Behaviour

Warehouse Management System (WMS) is a web-based application which follows MVC architecture by which WMS gets divided into three interconnected parts i.e. Model, View, and controller so that application can separate internal representation of information from each part. Here we have basic components of WMS:

1) Model – It specifies logical structure of data in WMS and high-level class associated with it. This layer is a domain-specific representation of data which describes working on an application. Whenever model changes its state, domain notifies its view so that they can refresh.
2) View – It is responsible to present all the UI logic of the application so that user can interact with the application. View renders model into a form which is suitable for interaction also multiple views can exist for a single model for different purposes. In WMS we have used HTML, CSS, jQuery, and AJAX to develop view.
3) Controller – It is a most important component of MVC architecture because it interfaces between Model and View components at the same time it processes all the business logic, incoming requests and manipulates data by using model component. It receives input from user process it with model and gives feedback to view component which is running in the client browser.

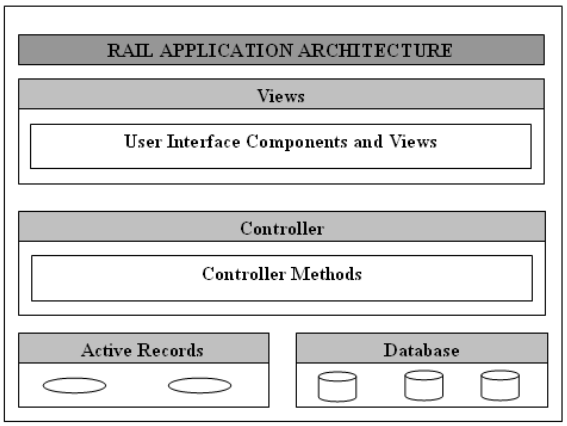

Figure 3.1 Layered architecture of rails application [17]

C. *Database Architecture*

Database architecture is set to specifications, rules, and standards which describes how data is stored in the database. It includes database tables, the data type of columns, relationship, associations and how they are working with each other. Database architecture includes anything which affects nature, structure, and flow of data which affects integrity, scalability, reliability, and performance of the application. Below figure shows the database architecture of WMS:

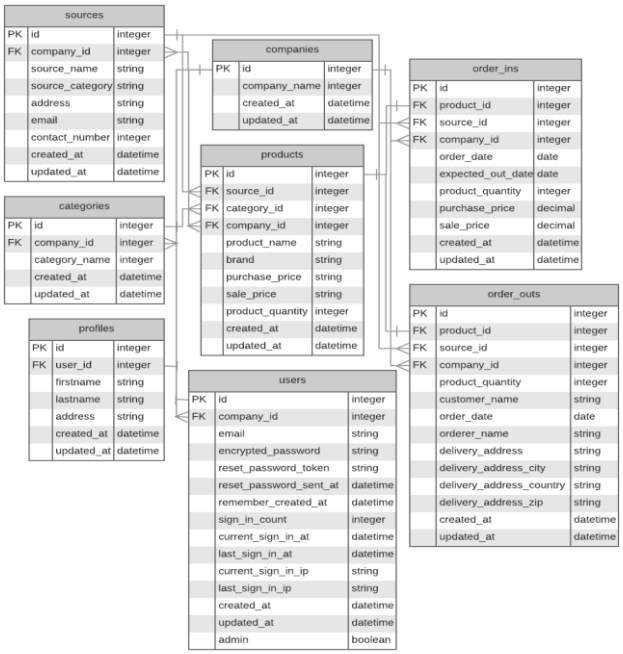

Figure 3.2 Database architecture of WMS [16]

D. *Modular Architecture*

The modular architecture of software development is a way to manage the complexity of requirement by breaking them down into smaller manageable modules, develop and test them independently. WMS have 2 complex modules as below:

1) Reports: WMS has the ability to generate various reports which are useful for users to perform their daily operations. Here is the basic architecture of reports module:

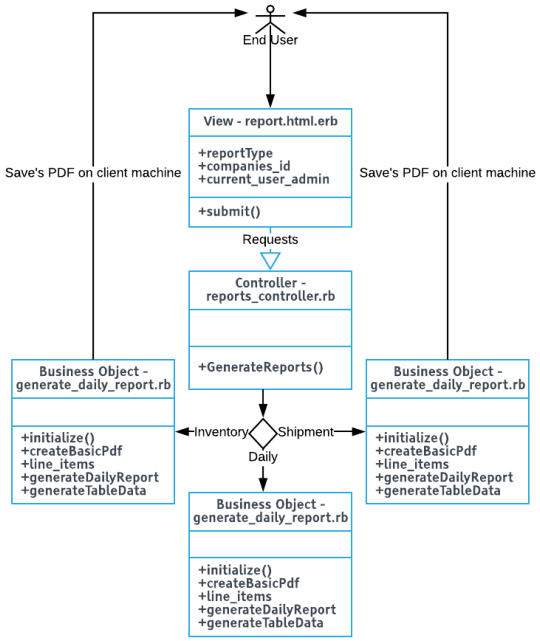

Figure 3.3 Database architecture of WMS [16]

2) Prediction System: In WMS admin user have the privilege prediction features by which he can predict sale of products in the specific geographical area so that he can give business ideas to individual warehouses:

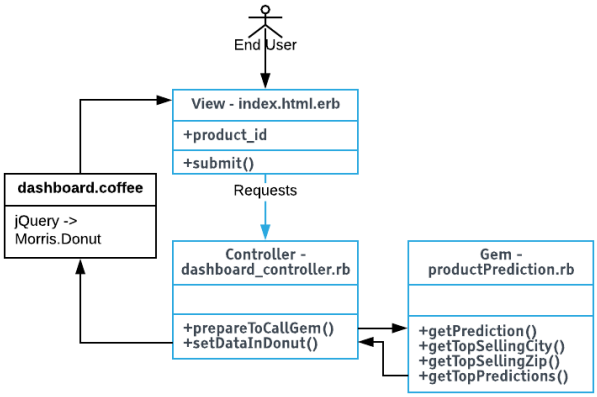

Figure 3.4 Database architecture of WMS [16]

IV. LIBRARY CREATION

Libraries are the collection of predefined modules which can add certain functionality to any project. ruby has a package manager to implement library called ruby gems. Rails framework make it very convenient to implement these gems into different application environments. This application makes full use of these gems to implement some functionality very easily which saves a lot of time and effort for the developers and allow to focus on developing other things

A. External gem

1) Devise – the application authentication, validation, encryption, session management and role-based authorization has been implemented by using this gem

2) rails-observers – this gem is used to implement observer design pattern to update the available product quantity for incoming and outgoing orders.

3) Pry-rails – this gem is used as a substitute for ruby IRB shell for debugging during development.

4) Prawn – it is used for WMS PDF generation system for the reporting functionality.

5) prawn-table – to implement pdf tables support for the reporting this gem is used.

6) bootstrap_sb_admin_base_v2 – this gem is used to provide the UI bootstrap template for the WMS system.

7) Jquery-rails – this gem is used to implement jQuery scripts in the application for the Bootstrap UI.

8) better_errors – during the development phase for a useful error page this gem replaces the rails standard error page and provides detailed information about the errors and the source of it.

9) brakeman - it is a rails security scanner used to analyze the application vulnerabilities and system flaws.

10) pg – The database for WMS production environment is Postgre.

11) morris-rails – it is used to integrate graphs generation for the prediction feature of the application

12) raphael-rails – this gem is used to support the implementation of morrisjs-rails to associate data with the

B. Implemented Gems

Ruby provides a standard format for distributing programs and libraries which can be used by other Ruby on Rails community. During development of WMS we need to create 2 gem files for different purposes as described below:

1) ProductPredictionWMS – This gem has been developed to predict the sale of the product in a geographical area in Ireland. WMS has been developed specially for the warehouses in Ireland which are having business in 4 major cities (Dublin, Cork, Galway, Limerick). This gem uses one of the popular data mining algorithm "Decision Tree" for giving a prediction of product. Although this may be true that ROR community already has gem developed for this algorithm, we have developed this gem for predicting sale of the product as per the zip code with weight to make a stronger prediction and gives only top 3 zip areas. This gem takes a 2-dimensional array as an input and gives hash as an output. Here is the example of input and output for this gem:

```
Input - [["Dublin", 1], ["Limerick", 2]  ,…]
Output - [["Dublin"],[["3", 4],["2", 4],["1", 3]]]
```

Input contains city name and zip code as parameters up to n numbers.
The output contains city name is present at first position and the second array contains zip code and strength of that zip for selling of the product.

2) DropdownSelection – This gem has been developed to dynamically update dropdown values of "zip code" drop-down in accordance with a drop-down for "city" in Ireland. WMS application has module "Order Out" where user enters product for delivery and user supposed to enter address of person who is placing order in that case user can select city by dropdown and application submits request with selected product id through AJAX call, gets values for zip code drop-down and fill dropdown of zip code by using jQuery. This gem is specially developed for cities in Ireland and we have restricted the zip code to 5 only because this is an example which can be expanded, and it is input data for prediction algorithm as well. Here is the example of input and output of gem:

```
Input - Dublin
Output - ["Dublin 1", "Dublin 2", "Dublin 3", …]
```

Input contains the name of the city.
The output contains zip code corresponding to cities.

V. IMPLEMENTATION

A software implementation is an act of transforming the detailed design into the valid program with the help of some programming languages and framework to produce a working application. The implementation phase involves not only writing programs but also debugging, compiling and construct the complete executable product. WMS has been implemented in "Ruby on Rails" framework with the help of programming languages explained below:

A. Front End

The presentation layer of WMS has been developed in HTML 5 which is standard markup language used to create web pages and web application, with CSS for styling and jQuery for client-side scripting of HTML [4] [5] [6]. WMS also built with AJAX, which allows a web application to change content dynamically without the need to reload the entire page [7]. Although this may be true that above technologies are under development current browsers supports most of its features. HTML 5 is much more elastic and implements a lot of new tags

and attributes but most importantly it makes user's browser independent from using third-party plugins.

. It is an abstract description of HTML along with various helpers which allow creating dynamic content. Haml greatly simplifies the generation of HTML creation even up to half of the efforts.

B. Back End

Database layer of WMS has the backbone of SQLite which is default database of the application when we create new Ruby on Rails project. We can easily change database of application with a single command, but we stick to SQLite because of its features such as in process library which implements serverless, self-contained, transactional SQL database engine [9]. When a user creates the basic structure of application autogenerated "config\database.yml" file contains complete configurations for SQLite database and important to realize that it has configurations for "development" as well as "production" environment.

The important reason of Ruby on Rails and many other frameworks supports SQLite 3 database as default for development is it can run in memory and backed by small files on disk which can easily create and move around. While easy to use, SQLite is not recommended to use in the production environment, so we are using PostgreSQL database as a service provided by Heroku which provides disk-backed storage [10].

C. Utilization of Plugins

With the most important feature of using rails i.e. using external plugins, we can extend functionality with a bunch of features which did not make it into core framework, and our application can use few of them as listed below:

1) sqlite3: This plugin allows software developers to interact with SQLite3 database engine so that he can perform basic operations such as select, update, insert and delete etc.

2) puma (3.7): It is a small library which provides a simple, fast, threaded, modern and concurrent web server for ruby's web-based applications [12]. Puma is default server for Ruby on Rails which is intended for use in development and production environment.

3) sass-rails (5.0): This library is a Sass adapter for Rails assets pipeline and provides official integration for Ruby on Rails projects with Sass stylesheet language [14].

4) uglifier (1.3.0): It minifies JavaScript files by wrapping UglifyJS to be accessible in Ruby [13].

5) coffee-rails (4.2): This gem provides CoffeeScript adapter for the rails assets pipeline with additional support to use CoffeeScript to respond JavaScript requests [15].

To simplify HTML creation process Haml has been used which is another markup language [8]

6) turbolinks (5): This library is useful for navigating in web application faster which use HTML to render application's views on the server side and link pages as usual. When the user follows the link, "turbolink" automatically fetches the page, swaps in its <body> and merges it's <head>, without waiting to load full page [17].

7) jbuilder (2.5): It provides software developer a simple digital subscriber line for declaring JSON structures. It is also helpful when generation process is intended to use conditional and loops.

D. Development and synchronization tool

Software development tools are a computer program which used by software developers to create, debug and maintain software applications. Tools may be discrete programs, executed separately often from the command line or maybe a single large program for especially for development and debugging called Integrated Development Environment (IDE). In the development process of WMS, we used below IDE's for development and debugging of application:

1) Cloud9: Cloud9 IDE is an online development environment which supports multiple programming languages such as Ruby, Python etc. It enables developers to start coding immediately with pre-configured workspaces and collaborate with team members with sharing workspace or collaborate coding features options.

2) Sublime text: It is cross-platform source code editor with python API supporting multiple programming as well as markup languages. It has some decent features such as go-to anything, multiple selections, separate command pallet and smart customizing option.

3) Git: It is a version control tool used for coordinating work on those files among multiple people.

E. Deployment

For deployment of WMS, we used "Heroku" which is a cloud Platform as a Service (PaaS) supporting multiple programming languages such as Ruby, Python, Java, Node.js etc. Cloud9 IDE used for development comes preconfigured to support Heroku deployment which saves the effort to configure the production environment. The application database needs to be migrated to Postgre during deployment and some manual entry was required in the database to ensure proper functioning of the application.

VI. DESIGN PATTERN

A design pattern is a reusable component that can be used to solve a recurring problem it contains the logic to solve a problem. Using design pattern are considered best practice for application development. some functionality of the application is implemented with the help of design pattern.

A. Product observer pattern:

This observer pattern is implemented with the help of rails-observer gem which provides an observable module. It gets triggered whenever a new incoming order is generated. The product observer updates the products quantity to the database and set the selling price of the product to the latest entry moreover it will update the purchase price of the product in the database by calculation the average of old and new prices.

B. Order out observer pattern:

This observer pattern is invoked whenever a user wants to create a new outgoing order it uses before_create function and check if the product quantity user want to deliver is available in inventory or not. If sufficient quantity is available to order is successfully generated and the quantity is automatically reduced in the product database however if the sufficient quantity is not available, it will not create the order and throw an error to the user.

C. Singleton:

Singleton design pattern can be used only when we want an only single instance of itself and provides global access to that instance i.e. that is it can be accessed it all parts of the application. During development of WMS, we came across a situation where the we want to debug application by using logger so we have implemented customized logger mechanism for WMS.

VII. TESTING

Test-driven development (TDD) is software development approach in which development is completely dependent on the very small development cycle. Here are few steps which get performed for TDD:

1) The software developer must clearly understand the requirement of applications.
2) Test cases written by the developer before any code is written for specific functionality.
3) Run the test and then verify the reason for failing the test case.
4) Write the code as a solution to pass that test which is as simple as possible which can be modified later.

During development of WMS we used existing rails test environment to run various testing methods such as model's unit testing, controllers functional testing, views testing and integration testing. Apart from the test-driven development approach, we have performed below levels of testing to ensure the quality of the product.

A. Unit Testing

It is a level of software testing where individual units/components of applications are tested to validate each unit of an application is performing as designed. In object-oriented programming, the smallest unit is a method which should be tested, and it can grow at the class level. A software developer who have written the code or his peer is responsible for unit testing. Unit testing alone cannot perform on a piece of the software application, but it can ensure that small units of code are working independently from each other.

B. Integration Testing

This is a type of software testing which seeks to verify the interfaces between components against software design. Software components may be integrated in the iterative way or all together, but recommended practice is to follow iterative way and perform integration testing after each iteration to ensure there are no defects in the interfaces and interaction between integrated components.

C. System Testing

System testing is software testing methodology which is conducted after complete integration of software component to evaluate system's performance with it's specified requirements. System testing is part of black box testing in which software tester examines the functionality of the application without knowing the internal structure of the application.

D. White Box Testing

This is an important method of testing the software in which software tester tests internal structure or functionality of the software. It is used at unit, integration and system level of testing to test control flow, data flow, and path testing.

E. Acceptance Testing

This is the final stage of testing a software application before moving to the production environment in which test conducted to determine if user requirements have been met or not. Acceptance testing is done by client side of an application and on another environment than development environment generally on their own hardware, is known as User Acceptance Testing (UAT). This may be performed as a part of hand-off process between two phases of software development and then tester of this process gives "go ahead" for deployment of an application in the production environment.

F. *Security Testing*

Security testing ensures that system and applications are free from any loopholes that may cause a big loss. In this testing software tester tests the application to find all the possible loopholes and weaknesses of applications which might result in loss of information at hands of end users or outsider. This involves multiple types such as vulnerability scanning, security scanning, penetration testing etc.

G. *Regression Testing*

This is a type of testing in which tester focuses on finding defects after a major code change has happened. Specifically, this seeks to uncover software regressions as degraded or lost features including old bugs. Such regression tests held whenever software functionality which was working fine is not working correctly at present time. A common method to do regression testing is to execute previously passed test cases and verify that previously fixed faults have re-emerged.

```
Finished in 7.455342s, 6.8407 runs/s, 6.1701 assertions/s.
51 runs, 46 assertions, 8 failures, 13 errors, 0 skips
```

Image 7.1 Screen capture of test results

VIII. PROJECT EVOLUTION

Warehouse Management System is Developed by Akash Hande and Pulkit Garg following the standard steps of software development and below is the contribution of each member.

A. Understanding Requirement Definition

Requirement gathering is the process of deeply understanding, analyzing, documenting, recording and taking interviews with stakeholder to understand their business requirements. During the development process of WMS requirement gathering is done by both the group members.

B. System and Software Design

Software application design is the process of implementing software solution with the help of entity relationship diagrams, flowcharts, database designs and class diagrams.

1) Database Design of WMS is done by both the team members together.

2) Complex features design – WMS have 2 complex features i.e. reports which were designed by Akash, Multi-warehouse designed by Pulkit.

C. Implementation and Unit Testing

The table below shows the task distribution amongst team members**.**

	Tasks	Pulkit	Akash
1	Scaffold creation and define the association between them	Order in, Product, Categories	Order out, Product, Sources
2	Integration of scaffolds and integration testing	Done	Done
3	Changed the presentation layer of the application with bootstrap		Done
4	Added authentication, created role-based security	Done	
5	Integrate application with newly developed UI and Authentication	Done	Done
6	Perform integration testing of application	Done	Done
7	A released first version of the application on Heroku	Done	Done
8	Developed dynamic feature – Reports, Multi-warehouse	Multi-warehouse	Reports
9	Implemented design pattern - Singleton, Observer	Order-ins observer	Order-outs observer, singleton
10	Integration of dynamic features and design patterns and perform integration testing	Done	Done
11	Developed gem files – ProductPrediction, DropdownSelection	DropdownSelection	ProductPrediction
12	Implemented gem files on the project and use them	DropdownSelection	ProductPrediction
13	Perform system testing as well as acceptance testing	Done	Done
14	Released final version of Heroku and perform regression testing	Done	Done
15	Prepare documentation for WMS	Done	Done

Table 8.1 workload distribution amongst teammates.

I. CONCLUSION AND FINDINGS

Warehouse management system has been developed for the medium and small industries having multiple warehouses in different locations to provide synchronized data among themselves at an affordable cost. The developed application can be used by different entities such as e-commerce companies, manufacturers, distributors and logistic suppliers for storage of their goods also for business analysis such as keeping an eye of inventory, predict sales of the product in specific geographical areas. The basic objective of WMS is to connect multiple warehouses located over the globe working under a single system and can have business with each other. Current application has been developed for warehouses in Ireland for demonstration purpose. The framework Ruby on Rails used for this application provides easy solutions that required minimum code to implement. Ruby on rails emphasize on convention over configuration and support quick application development by implementing external gems. Rails provide many functionalities out of the box and follow industrial standards. During development, it has been noticed that learning curve of ruby can be steep, and more time was spent on understanding the working of framework then development. While rails scaffold can generate most of the code it lacks the user interface. The IDE used for development of WMS is Cloud9 provides preconfigured workspaces which can be accessed from

anywhere, so developers can get more flexibility to develop an application.

While working with one of the sophisticated frameworks, in decent IDE we understood that there is still a scope for future enhancement of the application with below functionalities.

1) Barcode generation
   When the user is making incoming entry of any product in the warehouse the system should generate a barcode that can be used to track that product with its unique characteristic. "Barby" is a gem file which is already developed for generating barcodes which can be used for this purpose[1].
2) Easy registration
   When the user is registering a product for delivery/orders out, he should be able to register it by using barcode. "ruby-barcodescanner" is a gem which can be used for this purpose [2].
3) Easy Tracking:
   We can add "Rack Suggestion" algorithm in accordance with "Dijkstra's Algorithm" which will give a shortest possible path to reach rack suggested by system [3].
4) Dynamic environment
   Currently, WMS has only 3 warehouses, but it is developed with the intention that Admin can dynamically register more warehouses across the globe.

   Further, we can also provide dynamic reporting functionality such as auto-generated email for reports, specific for warehouses etc. which will enable users to perform their daily operations with more ease and eventually some speed in their daily tasks.

   The dashboard can be more dynamic with the features such as monthly/weekly/daily sale of the product, more predictive for sale, business for the same interest of warehouses etc.

In the final analysis, we have discovered that ruby on rails is a very convenient framework to develop, test and deploy cloud-based software application where developer do not have to worry on security aspects, integration of external tools and libraries.

Bibliography

[1]"Chapter 1: What is Software Architecture?", *Msdn.microsoft.com*, 2018. [Online]. Available: https://msdn.microsoft.com/en-us/library/ee658098.aspx.

[2]"Framework Definition", *Techterms.com*, 2018. [Online]. Available: https://techterms.com/definition/framework.

[3]"Ruby on Rails MVC Framework", *www.tutorialspoint.com*, 2018. [Online]. Available: https://www.tutorialspoint.com/ruby-on-rails/rails-framework.htm.

[4]"HTML 5.2", *W3.org*, 2018. [Online]. Available: https://www.w3.org/TR/html5/.

[5]"Cascading Style Sheets", *W3.org*, 2018. [Online]. Available: https://www.w3.org/Style/CSS/Overview.en.html.

[6]j. jquery.org, "jQuery", *Jquery.com*, 2018. [Online]. Available: https://jquery.com/.

[7] j. jquery.org, "Ajax | jQuery API Documentation", *Api.jquery.com*, 2018. [Online]. Available: https://api.jquery.com/category/ajax/.

[8]"Haml", *Haml.info*, 2018. [Online]. Available: http://haml.info/.

[9]"SQLite Home Page", *Sqlite.org*, 2018. [Online]. Available: https://www.sqlite.org/index.html.

[10]"Adding Authentication with Devise - Rails Girls", *Guides.railsgirls.com*, 2018. [Online]. Available: https://guides.railsgirls.com/devise.

[11] J. http://jordanisip.com, "A Modern, Concurrent Web Server for Ruby - Puma", *Puma.io*, 2018. [Online]. Available: http://puma.io/.

[12]"uglifier | RubyGems.org | your community gem host", *Rubygems.org*, 2018. [Online]. Available: https://rubygems.org/gems/uglifier/versions/4.1.6.

[13]2. May 21, "An Introduction to Sass in Rails — SitePoint", *SitePoint*, 2018. [Online]. Available: https://www.sitepoint.com/an-introduction-to-sass-in-rails/.

[14]"CoffeeScript", *Coffeescript.org*, 2018. [Online]. Available: http://coffeescript.org/.

[15]"Pry - an IRB alternative and runtime developer console", *Pryrepl.org*, 2018. [Online]. Available: http://pryrepl.org/.

[16]"Online Diagram Software & Visual Solution | Lucidchart", *Lucidchart*, 2018. [Online]. Available: https://www.lucidchart.com/.

[17]"Ruby on Rails MVC Framework", *www.tutorialspoint.com*, 2018. [Online]. Available: https://www.tutorialspoint.com/ruby-on-rails/rails-framework.