GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Tickling

A Project Report of Project - 2

Submitted by

TAPAN KUMAR RAJPUT

(1713104001)

in partial fulfilment for the award of the degree of

BACHELOR OF COMPUTER APPLICATION(BCA)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

MR. S SREEJI
Professor

APRIL / MAY- 2020

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **" TICKING"** is the bonafide work of "***TAPAN KR. RAJPUT* (1713104001)"** who carried out the project work under my supervision.

SIGNATURE OF HEAD
Dr. MUNISH SHABARWAL
PhD (Management), PhD (CS)
Professor & Dean,
School of Computing Science & Engineering

SIGNATURE Of SUPERVISOR
MR.S SREEJI
Professor
School of Computing Science & Engineering

Table of Contents

Chapter **Pages**

Chapter **Pages**

1. Introduction

❖ General

Tickling is a web based ticket management system. The purpose of a ticket management system is to keep the track of tasks and monitor their implementation. In each organization, Directors, Managers, Team Leaders, Supervisors, etc need to assign task to their subordinates. Most of the time implementation of tasks fails or lags behind the schedule because there exists no record of the assigned task and no mechanism to monitor it**. Tickling** proposes to solve this problem, by keeping the track of each assigned task, by supporting different priority levels of the tasks, by supporting notifications and alerts of pending and to be monitored tasks for assignee and assignor respectively, and by supporting daily, weekly, monthly and quarterly reports for both assignor and assignee.

It is proposed to be designed as a product which can be customized according to the requirements of the clients. The company proposes to make this product available to all the SMEs, Academic institutions, NGOs and even political parties.

The product initially will have web based interface which will be complemented by android and ios based apps later. To fulfill this need, service oriented architecture (SOA) is used in the implementation of the product.

Major Modules of the product are:-

- User Manager
- Tasks Manager
- Notification Manager
- Reporting Manager
- Customization Manager

❖ Objective of the project

- The main purpose of our online ticket management system is toprovide another way for the customer to buy product. It isan automatic system.

- After inserting the data to database,staff need not to due withthe order receive through the system. In fact, there is similar system on the internet, but there is no refund method found inthe existing system.

- This system is basically aimed to provide the customer thecomplete information of the product, according to which thecustomer can book the products and the refund facility providesmore flexibility to the system.

❖ The goals of our system are:

- To provide an anytime anyplace service for the customer.
- To minimize the number of staff.
- To promote the products on the internet.

2. Feasibility Study

Feasibility studies are crucial during the early development of any project and form a vital component in the development process. A feasibility study is a management-oriented activity. It is a test of system proposal according to its workability, impact on the organization, ability to meet user needs and effective use of resources. A feasibility study is conducted to select the best system that meets performance requirement. This entails an identification description, an evaluation of candidate system and the selection of best system for the job. The system required performance is defined by a statement of constraints, the identification of specific system objective and a description of outputs.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities.

The key considerations in feasibility analysis are as follows:

2.1.1 Economic Feasibility

2.1.2 Technical Feasibility

2.1.3 Operational Feasibility

Economic feasibility:

It looks at the financial aspects of the project. It determines whether the management has enough resources and budget to invest in the proposed system and the estimated time for the recovery of cost incurred. It also determines whether it is worthwhile to invest the money in the proposed project. Economic feasibility is determines by the means of cost benefit analysis. The proposed system is economically feasible because the cost involved in purchasing the hardware and the software are within approach. The personal cost like salaries of employees hired are also nominal, because working in this system need not required too many professional. The operating-environment costs are marginal. The end user need not to require paying for the membership to the portal hence the application is pity economical in comparison to the traditional approach. Users of this portal have to pay minimal in the terms of internet charges. The software used in this project are Java as front end which is open source, similarly the backend required for storing details is also the same database that is My SQL which is open source. So the project is economically feasible in terms of development and use.

Technical Feasibility:

It is a measure of the practically of a specific technical solution and the availability of technical resources and expertise.

- The proposed system uses Java as front-end and My Sql as back-end tool.
- Java with J2EE edition help in easy development of website by the help of Servlet and Jsp.
- My sql is a popular tool used to design and develop database objects such as table views, indexes.

- The above tools are readily available, easy to work with and widely used for developing commercial application.

As the whole project is web based application internet connectivity is required which is readily available to the individuals. In the terms of software the project is built on java which is open source as front end and My SQL as backend and computers with browser enable facility is required which is too readily available to the many of the operating system.

Hardware used in this project are- p4 processor 2.4GHz and above, 1 GB RAM, 200 GB hard disk. These hardware were already available on the existing computer system. The software like My Sql, tomcat Server, Thin Driver, JDK, JSDK, J2EE and operating system WINDOWS-XP' used were already installed On the existing computer system. So no additional hardware and software were required to purchase and it is technically feasible. The technical feasibility is in employing computers to the organization. The organization is equipped with enough computers so that it is easier for updating. Hence the organization has not technical difficulty in adding this system.

Operational Feasibility:

The system will be used if it is developed well then tackle the hindrances in its operation.

- No major training and new skills are required as it is based on DBMS model.
- It will help in the time saving and fast processing and dispersal of user request and applications.
- New product will provide all the benefits of present system with better performance.
- Improved information, better management and collection of the relevant data which is very helpful to users of the project.
- Multi User support facilitate in smooth running of software.
- User will have control over own information. Important information such as mail address cannot be publicly viewed.
- Faster and systematic processing of user application approval, view acceptance and rejection facility.

Hence the project will be operationally feasible also.

Behavioral Feasibility:

In this type of feasibility check, we come to know if the newly developed system will be taken and accepted by the working force i.e. the people who will use it.

As the application has graphical user interface with suitable screen which help them a lot in navigation through the pages without concern of the backend processes. So the users must not have any problem in finding or searching the content from the site. Hence results in behaviorally feasible project.

3. Software Requirement Specification

❖ Introduction:

The System Requirement Specification includes a description of every input into the system, every output from the system and all functions performed by the system in response to input or in support of an output. In addition it also contains non-functional requirements. Non-functional requirements impose constraints on the design or implementation such as performance engineering requirements, quality standards, or design constraints.

The SRS may be one of a contract deliverable Data Item Descriptions. It is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. The software requirements specification document enlists all necessary requirements that are required for the project development. An SRS document specifies the required behavior the system in terms of input data, required processing, output data, operational scenarios and interfaces and the attributes of a system including performance, security, maintainability, reliability, audit ability, availability and safety requirements and design constraints. Alias: user requirement specification, functional specification.

To derive the requirements we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with the project team and customer.

(a). Purpose:

The purpose of this project is to present a detailed description of the product and booking of the product for buying online. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

(b).Interface:

I. Software Interface:

Software specification:

JAVA

Java is a small, simple, safe, object oriented, interpreted or dynamically optimized, byte coded, architectural, garbage collected, multithreaded programming language with a strongly typed exception-handling for writing distributed and dynamically extensible programs.

Java is an object oriented programming language. Java is a high-level, third generation language like C, FORTRAN, Small talk, Pearl and many others. You can use

Java to write computer applications that crunch numbers, process words, play games, store data or do any of the thousands of other things computer software can do.

Special programs called applets that can be downloaded from the internet and played safely within a web browser. Java a supports this application and the follow features make it one of the best programming languages.

- It is simple and object oriented
- It helps to create user friendly interfaces.
- It is very dynamic.
- It supports multithreading.
- It is platform independent
- It is highly secure and robust.
- It supports internet programming

Java is a programming language originally developed by Sun Microsystems and released in 1995 as a core component of Sun's Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code which can run on any Java virtual machine (JVM) regardless of computer architecture.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the Java

Community Process, Sun made available most of their Java technologies as free software under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java and GNU Class path.

The Java language was created by James Gosling in June 1991 for use in a set top box project. The language was initially called *Oak*, after an oak tree that stood outside Gosling's office - and also went by the name *Green* - and ended up later being renamed to *Java*, from a list of random words. Gosling's goals were to implement a virtual machine and a language that had a familiar C/C++ style of notation.

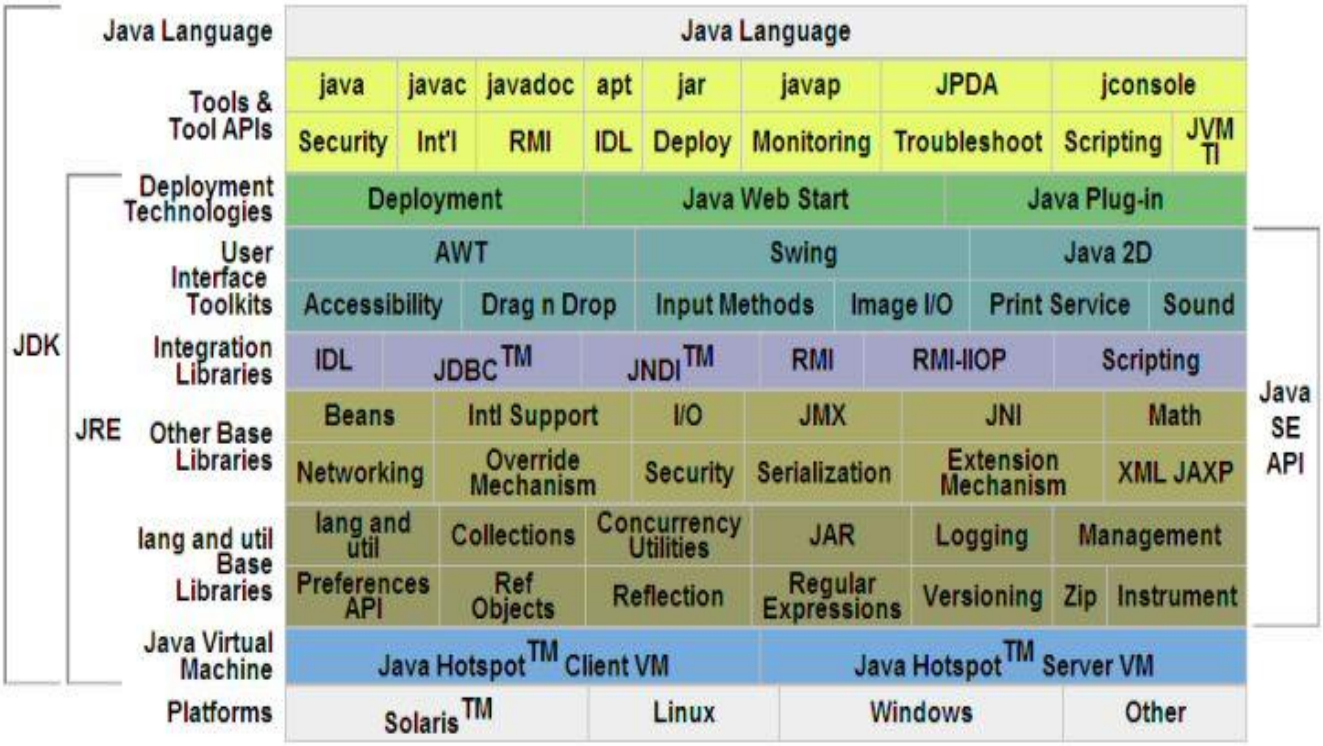
Primary goals:

There were five primary goals in the creation of the Java language:

1. It should use the object-oriented programming methodology.
2. It should allow the same program to be executed on multiple operating systems.
3. It should contain built-in support for using computer networks.
4. It should be designed to execute code from remote sources securely.
5. It should be easy to use by selecting what were considered the good parts of other object-oriented languages.

The Java Platform consists of several programs, each of which provides a distinct portion of its overall capabilities. For example, the Java compiler, which converts Java source code into Java byte code (an intermediate language for the Java Virtual Machine (JVM)), is provided as part of the Java Development Kit (JDK). The sophisticated Java Runtime Environment (JRE), complementing the JVM with a just-in-time (JIT) compiler, converts intermediate byte code into native machine code on the fly. Also supplied are extensive libraries (pre-compiled into Java byte code) containing reusable code, as well as numerous ways for Java applications to be deployed, including being embedded in a web page as an applet. There are several other components, some available only in certain editions.

The essential components in the platform are the Java language compiler, the libraries, and the runtime environment in which Java intermediate byte code "executes" according to the rules laid out in the virtual machine specification.



JAVA VIRTUAL MACHINE:

The heart of the Java Platform is the concept of a "virtual machine" that executes Java byte code programs. This byte code is the same no matter what hardware or operating system the program is running under. There is a JIT compiler within the *Java Virtual Machine*, or JVM. The JIT compiler translates the Java byte code into native processor instructions at run-time and caches the native code in memory during execution.

The use of byte code as an intermediate language permits Java programs to run on any platform that has a virtual machine available. The use of a JIT compiler means that Java applications, after a short delay during loading and once they have "warmed up" by being all or mostly JIT-compiled, tend to run about as fast as native programs. Since JRE version 1.2, Sun's JVM implementation has included a just-in-time compiler instead of an interpreter.

Although Java programs are Platform Independent, the codes of the Java Virtual Machine (JVM) that execute these programs are not. Every Operating System has its own JVM.

JAVA RUNTIME ENVIRONMENT:

The Java Runtime Environment, or *JRE*, is the software required to run any application deployed on the Java Platform. End-users commonly use a JRE in software packages and Web browser plugins. Sun also distributes a superset of the JRE called the Java 2 SDK (more commonly known as the JDK), which includes development tools such as the Java compiler, Javadoc, Jar and debugger.

One of the unique advantages of the concept of a runtime engine is that errors (exceptions) should not 'crash' the system. Moreover, in runtime engine environments such as Java there exist tools that attach to the runtime engine and every time that an exception of interest occurs they record debugging information that existed in memory at the time the exception was thrown (stack and heap values). These Automated Exception Handling tools provide 'root-cause' information for exceptions in Java programs that run in production, testing or development environments.

WEB COMPONENT:

J2EE Web components can be either servlets or JSP pages. *Servlets* are Java programming language classes that dynamically process requests and construct responses. *JSP pages* are text-based documents that execute as servlets but allow a more natural approach to creating static content.Static HTML pages and applets are bundled with Web components during application assembly, but are not considered Web components by the J2EE specification. Server-side utility classes can also be bundled with Web components and, like HTML pages, are not considered Web components.

JAVA SERVER PAGE (JSP):

Java Server Pages technology is the Java platform technology for building applications containing dynamic Web content such as HTML, DHTML and XML. The Java Server Pages technology enables the authoring of Web pages that create dynamic content easily but with maximum power and flexibility.

The Java Server Pages technology offers a number of advantages:

- Write Once, Run Anywhere properties:

  The Java Server Pages technology is platform independent, both in its dynamic Web pages, its Web servers, and its underlying server components. You can author JSP pages on any platform, run them on any Web server or Web enabled application server, and access them from any Web browser. You can also build the server components on any platform and run them on any server.

- High quality tool support:

  The Write Once, Run Anywhere properties of JSP allows the user to choose best-of-breed tools. Additionally, an explicit goal of the Java Server Pages design is to

enable the creation of high quality portable tools.

- Reuse of components and tag libraries:

The Java Server Pages technology emphasizes the use of reusable components such as: JavaBeans components, Enterprise JavaBeans components and tag libraries. These components can be used in interactive tools for component development and page composition. This saves considerable development time while giving the cross-platform power and flexibility of the Java programming language and other scripting languages.

- Separation of dynamic and static content:

The Java Server Pages technology enables the separation of static content from dynamic content that is inserted into the static template. This greatly simplifies the creation of content. This separation is supported by beans specifically designed for the interaction with server-side objects.

- Support for scripting and actions:

The Java Server Pages technology supports scripting elements as well as actions. Actions permit the *encapsulation* of useful functionality in a convenient form that can also be manipulated by tools; scripts provide a mechanism to *glue together* this functionality in a per-page manner.

JSP ARCHITECTURE:

JSPs are built on top of SUN's servlet technology. JSPs are essential an HTML page with special JSP tags embedded. These JSP tags can contain Java code. The JSP file extension is .jsp rather than .htm or .html. The JSP engine parses the .jsp and creates a Java servlet source file. It then compiles the source file into a class file; this is done the first time and this why the JSP is probably slower the first time it is accessed. Any time after this the special compiled servlet is executed and is therefore returns faster.

Software Requirements:

For development:

Browser	:	Any java enabled browser
IDE	:	My Eclipse8.5
Database	:	MySql
Operating System	:	Window xp and above

Web server	:	Apache Tomcat 7.x
Documentation tool	:	Microsoft office
Scripting language:		JavaScript

Server:

Browser	:	Any java enabled browser
Database	:	MySql
Web server	:	Apache Tomcat 7
Operating System	:	Any O.S. (windows/Linux/etc)

Client:

Browser	:	Any browser
Operating System	:	Any O.S. (windows/Linux/etc)

II. Hardware Interface:

Hardware Requirements:

For development:

Main Processor	:	Core I3 and above.
Hard-disk Capacity	:	120 GB
RAM	:	3 GB

For Client

Main Pr0ocessor	:	dual core and above
Hard-disk Capacity	:	80GB
RAM	:	1 GB

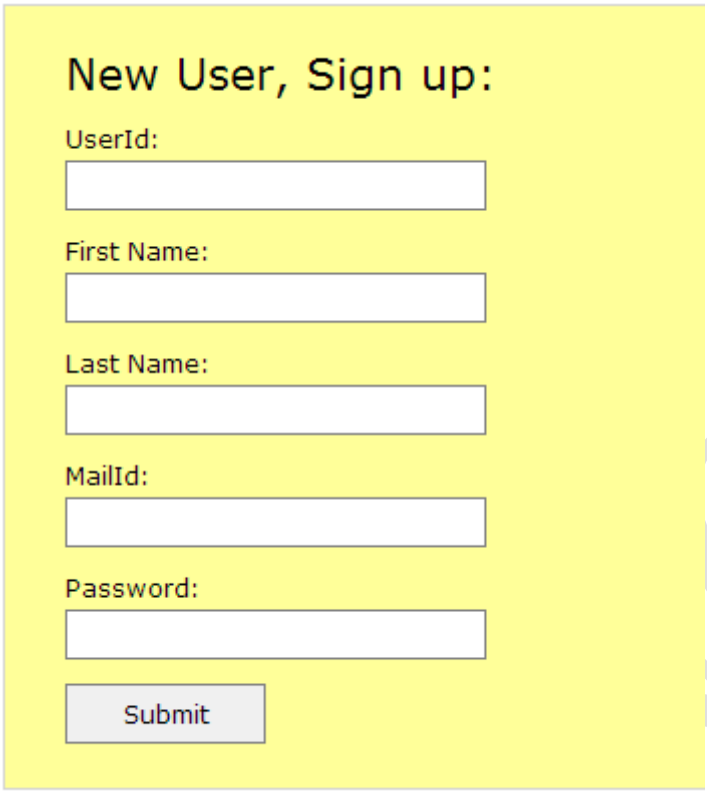
For Server

Main Processor	:	Core I3 and above.
Hard-disk Capacity	:	120 GB
RAM	:	3 GB

III. User Interface:

User Interfaces:

After the application deployed the product will have home page which have a proper facility for sign in and sign up. In case the user is not registered yet, he can enter the registration form after clicking on “sign up” link as shown in below figure.



Similarly the smooth Interface should be provided. For The more elaboration snap shot are attached at the end.

Acronyms and Abbreviations:

Word	Meaning
Tickling	Online Ticket Management System
JSP	Java Server pages
IEEE	Institute of Electrical and Electronics Engineers
Jdk1.6	Java development kit version 1.6

4. Design

4.1. Introduction:

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design

i. LOGICAL DESIGN:

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Reviews the current physical system – its data flows, file content, volumes, Frequencies etc.
- Prepares output specifications – that is, determines the format, content and Frequency of reports.
- Prepares input specifications – format, content and most of the input functions.
- Prepares edit, security and control specifications.
- Specifies the implementation plan.
- Prepares a logical design walk through of the information flow, output, input, Controls and implementation plan.
- Reviews benefits, costs, target dates and system constraints.

ii. PHYSICAL DESIGN:

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

- Design the physical system.
- Specify input and output media.
- Design the database and specify backup procedures.
- Design physical information flow through the system and a physical design.
- Plan system implementation.
- Prepare a conversion schedule and target date.
- Determine training procedures, courses and timetable.
- Devise a test and implementation plan and specify any new hardware/software.
- Update benefits, costs, conversion date and system constraints.

4.2. System Design:

Software development approaches

System development is step by step process which constitutes phases for the successful deployment of the project. Each phase should be properly carried out to make the project run in sound way. From the initial requirement to implementation phase it should be well managed. Below figure illustrates the normally included phase in project development and is also known as system development life cycle (SDLC).

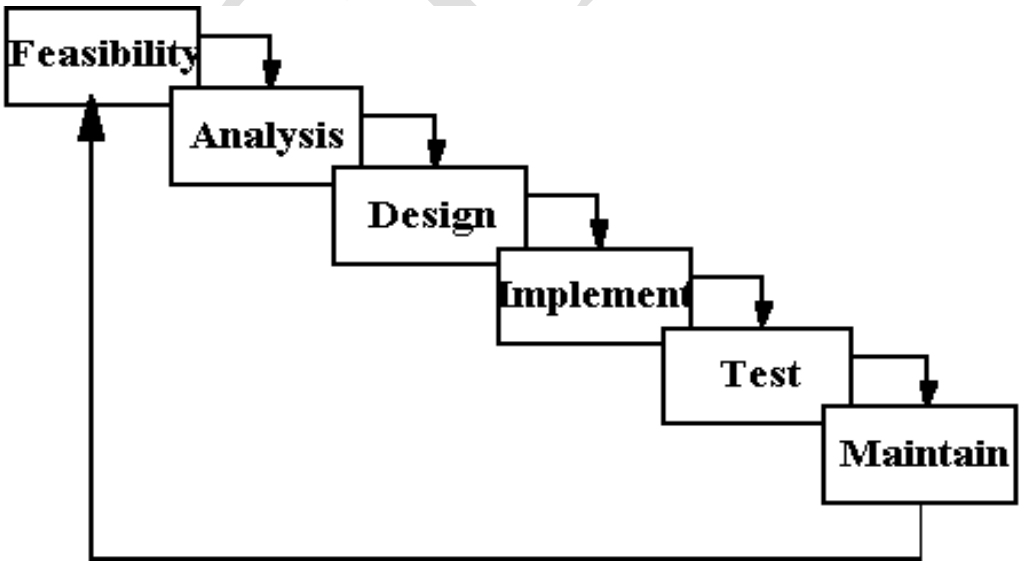


Fig.-System Development Life Cycle (SLDC).

4.5. Data Structures:

(1) **UserMaster:** This table shall store information of users. A tickling user can have one of the following roles: Admin, Assignor, Assignee. Admin user will be responsible for customization of the product, and creating & managing assignor and assignee accounts. Assignor user will be responsible for assigning tasks, monitoring their progress, sending notification and alerts and viewing reports. Assignee user will be responsible for viewing the assigned jobs, completing them, viewing notification and reports. UserMaster table will have following fields.

Field name	Data type	Constraint/Description
userId	Number	Primary key, (Auto Generated)
Username	Varchar	Not Null
userTitle	Varchar	Not Null (Job title of the user e.g. Manager, CEO, Director etc)
mailId	Varchar	Will be used as loginId.
Password	Varchar	Not Null
createdOn	Date	User account creation date
Status	Integer	Active, Archived (Status of user accounts can be changed by the admin user)
archivedOn	Date	Date on which account is archieved
archiveReason	Integer	Left job, transferred, suspended etc (all the reasons will be represented numerically)
productId	Number	Foreign Key (ProductMaster)

(2) **UserRoles:** This table shall be used to store the information of roles of users. A separate table is required because a user may be assigned different roles e.g. A manager can be both assignor as well as assignee, he may assign tasks to his subordinates but can be assigned task by the director or CEO. It contains following fields:

Field name	Data type	Constraint/Description
userId	Number	Foreign key (UserMaster)
Role	Number	Admin, Assignor, Assignee (will be numerically represented)

(3) **Assignor_Assignee:** This table shall store information of assignees to whom an assignor can assign tasks. It contains following fields.

Field name	Data type	Constraint/Description
assignorId	Number	Foreign key (UserMaster)
assigneeId	Number	Foreign key (UserMaster)
allotmentDate	Date	Date on which assignee is alloted to the assignor
revocationDate	Date	Date on which assignee is revoked for the assignor
assingeeStatus	Number	(Active, Revoked) is used to determined whether task can be assigned to the assignee or not.

(4) **LoginMaster:** This table shall store login information of the registered users. It contains following fields:

Field name	Data type	Constraint/Description
userId	Number	Foreign key(UserMaster)
lastLoginDate	Date	Not Null
loginMode	Number	(web, andorid app, ios app etc)

(5) **TicketMaster:** This table shall store the information of the tasks which will be assigned to different assignees by assignors. It contains following fields.

Field name	Data type	Constraint/Description
ticketId	Number	Primary key (Auto Generated)
ticketTitle	Varchar	Not Null (max 100 chars)
ticketDesc	Varchar	Max 2000 chars
Priority	Number	(Normal, Medium, Urgent)
assignedBy	Number	Foreign Key (UserMaster)
assignedTo	Number	Foreign Key (UserMaster)
assignedOn	Date	Not Null

completionDate	Date	Not Null
Status	Number	(Open, Closed, Active, Pending)
has Attachments	Boolean	A ticket may have text, pdf, images, audio or video files as attachment
hasReplies	Boolean	A ticket may have a set of replies representing the conversation b/w the assignor and assignee.

(6) **TicketAttachments:** This table contains the information of files which are submitted as attachment of a ticket. It contains following fields.

Field name	Data type	Constraint/Description
ticketId	Number	Foreign key (TicketMaster)
Filename	Varchar	Name of the attached file.
filePath	Varchar	Path of the file on the server.

(7) **TicketReplies:** This table contains information of the replies exchanged by the assignor and assignee on a ticket. It contains following fields.

Field name	Data type	Constraint/Description
replyId	Number	Primary key (Auto Generated)
ticketId	Number	Foreign key (TicketMaster)
Replier	Varchar	Foreign key (UserMaster)
replyDate	DateAndTime	Not Null
replyTitle	Varchar	Title of the reply
replyDesc	Varchar	Description of the reply
has Attachments	Boolean	A reply may also have attachments.

(8) **ReplyAttachments:** This table contains the information of files which are submitted as attachment of a reply. It contains following fields.

Field name	Data type	Constraint/Description

replyId	Number	Foreign key (TicketReplies)
Filename	Varchar	Name of the attached file.
filePath	Varchar	Path of the file on the server.

(9) **NotificationMaster:** This table contains the information of notifications which will be sent by tickling system to the assignor and assignees. It contains following fields.

Field name	Data type	Constraint/Description
notificationId	Number	Primary key (Auto Generated)
Message	Varchar	Notification message.
notificationType	Number	(Ticket Opened, Closed, Pending Tickets, New assignee added, assignee revoked etc)
notificationDate	DateAndTime	Not Null
sentTo	Number	userId of userMaster as Foreign Key

(10) **ProductMaster:** This table contains the company specific information used by an administrator to customize the tickling product. It contains following fields.

Field name	Data type	Constraint/Description
productId	Number	Primary Key (Auto Generate)
companyName	Varchar	Company name to be displayed on the product.
companyLogo	Varchar	Path of logo image of the company.
Tagline	Varchar	Tag line of the company if any.
websiteURL	Varchar	URL of company web site.
productURL	Varchar	Product URL for back linking to the company website of the customer.

4.6. Description of Modules:

The Tickling product contains following modules:

- User Manager
- Tasks Manager
- Notification Manager
- Reporting Manager
- Customization Manager

User Manager: This Module will be used by all type of users i.e. admin, assignor and assignee. Following functionality will be provided by this module: registration, login, logout, view and edit profile, forgot password, change password, assigning roles to users, allocating and revoking assignees, activating and archiving user accounts.

Tasks Manager: This module provides functionality of managing tasks. Using it an assignor can assign tasks, can view pending and completed tasks, can send and view reply and attachments as part of task. An assignee can view assigned tasks, can reply to them, and can close them using this module.

Notification Manager: This module provides functionality of sending and viewing notifications. With the help of this module different notification related to new task, task completion, pending tasks, account creation, deactivation etc will be sent to different type of users.

Reporting Manager: This module will be responsible for generating user wise, date wise, task wise reports.

Customization Manager: This module will be used by the administrator of a company to customize the tickling product according to the requirements of the company, and to generate company specific product URL so that it can be integrated to the company' web site.

4.7. SnapShot:

Home Page:

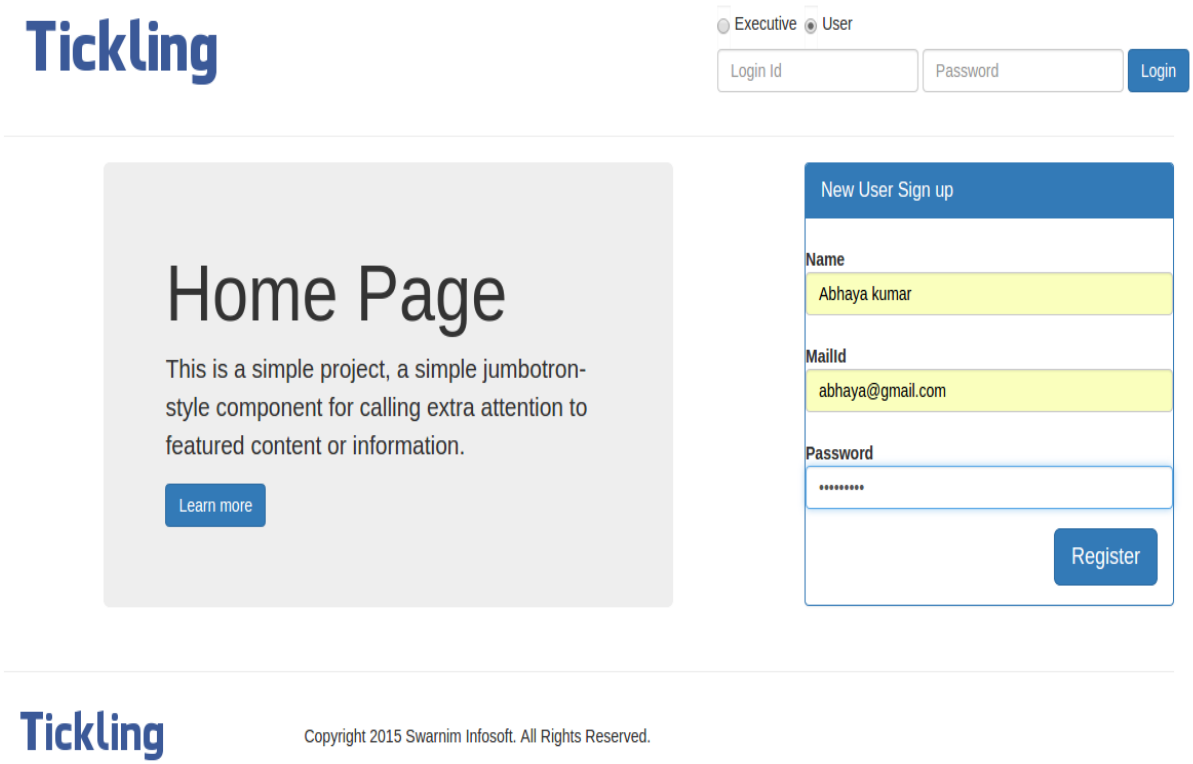


Fig:-Home Screen for Signup and Login

Welcome Page:

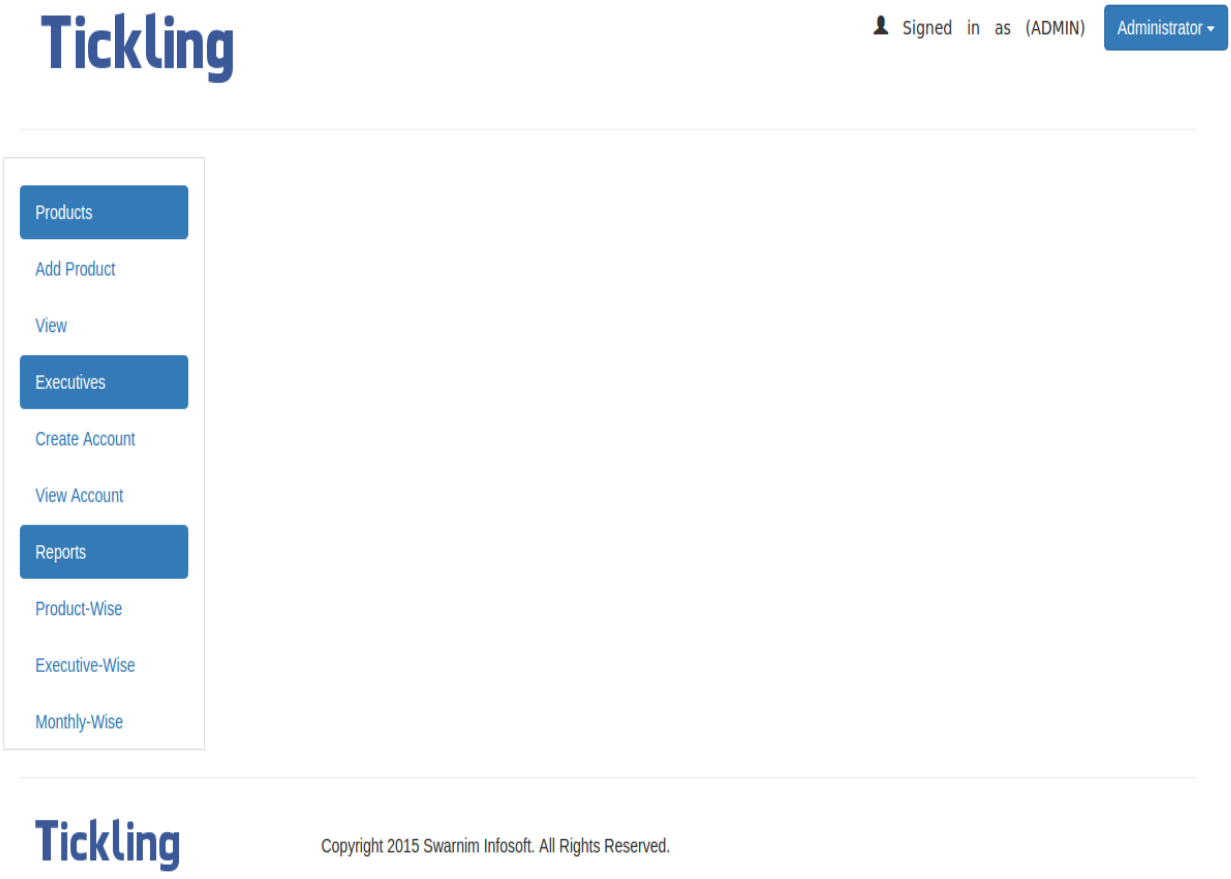


Fig:-Welcome Screen for Admin.

Add Product Page:

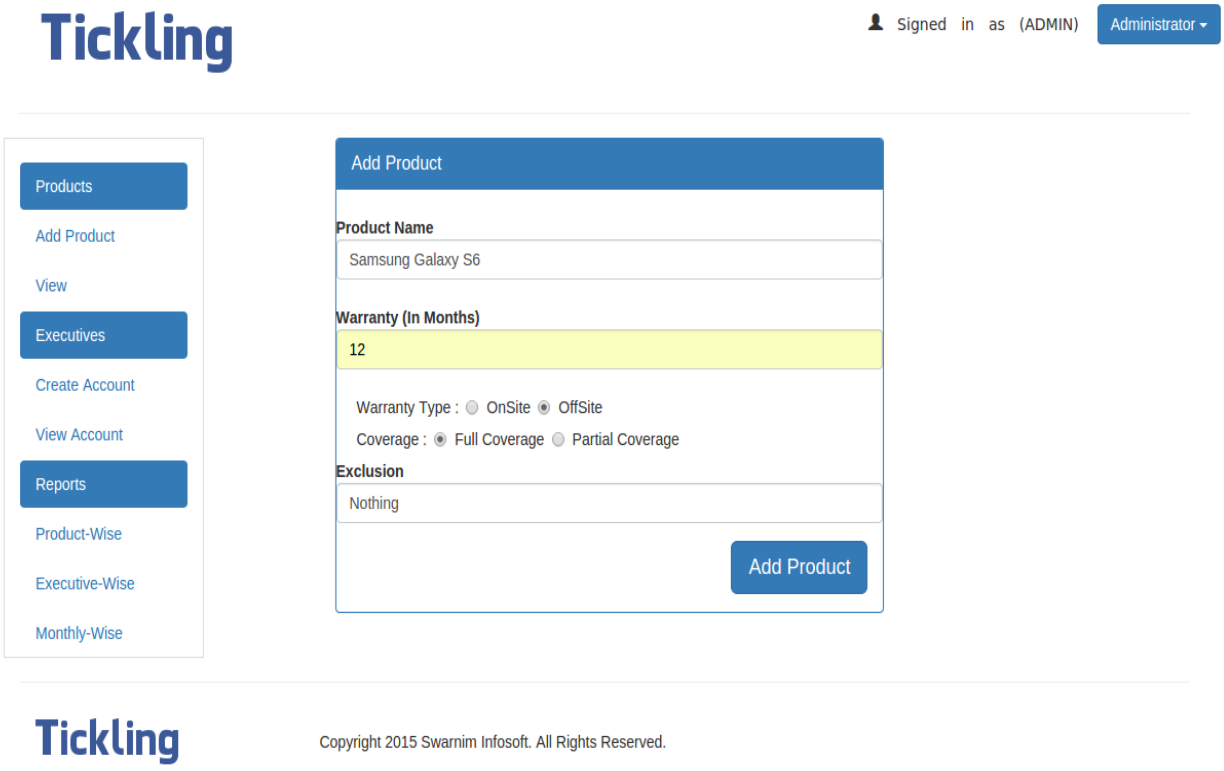


Fig:- Add Product Page By Admin

Product View Page:

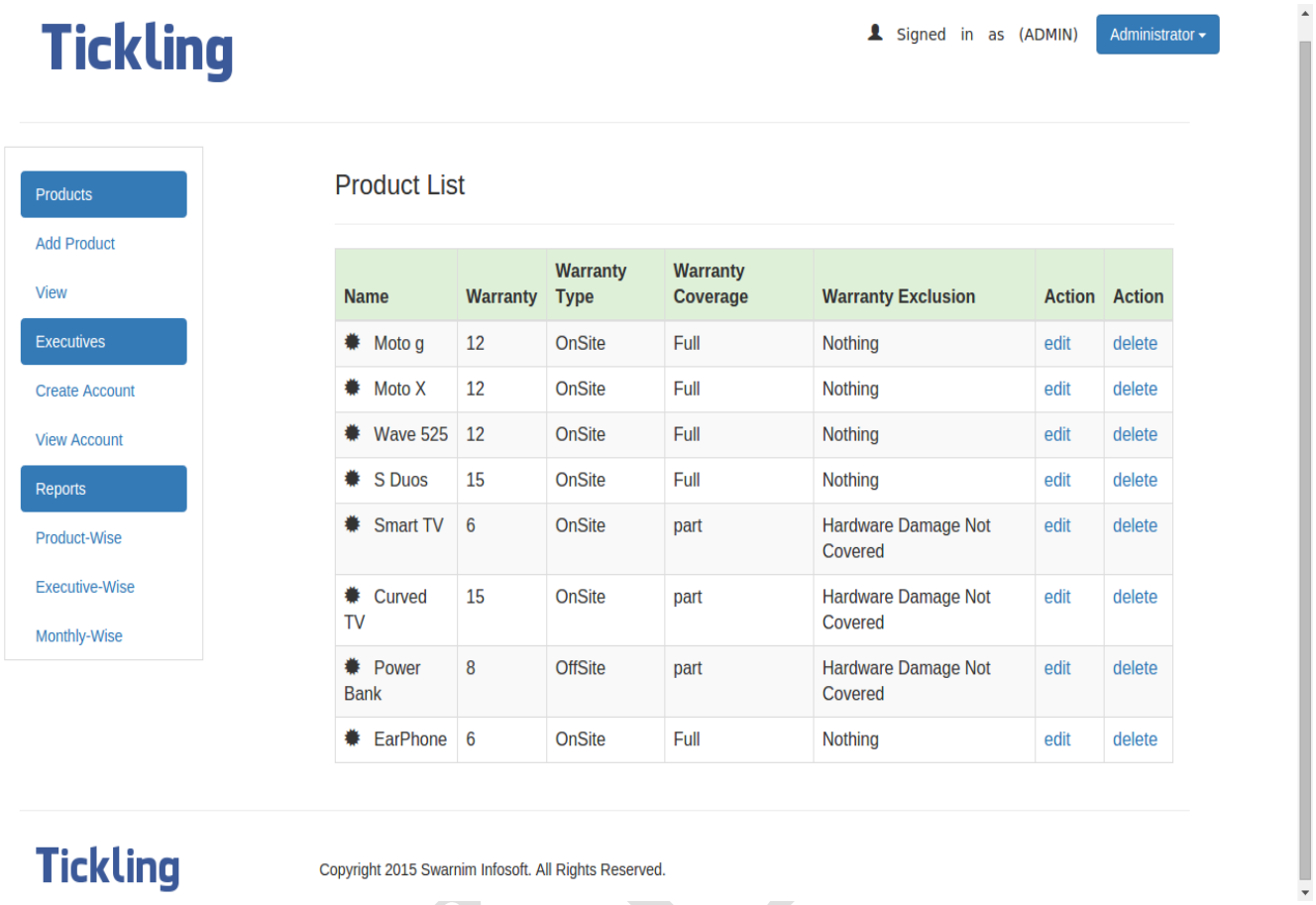


Fig:- Product View.

Edit Product Details:

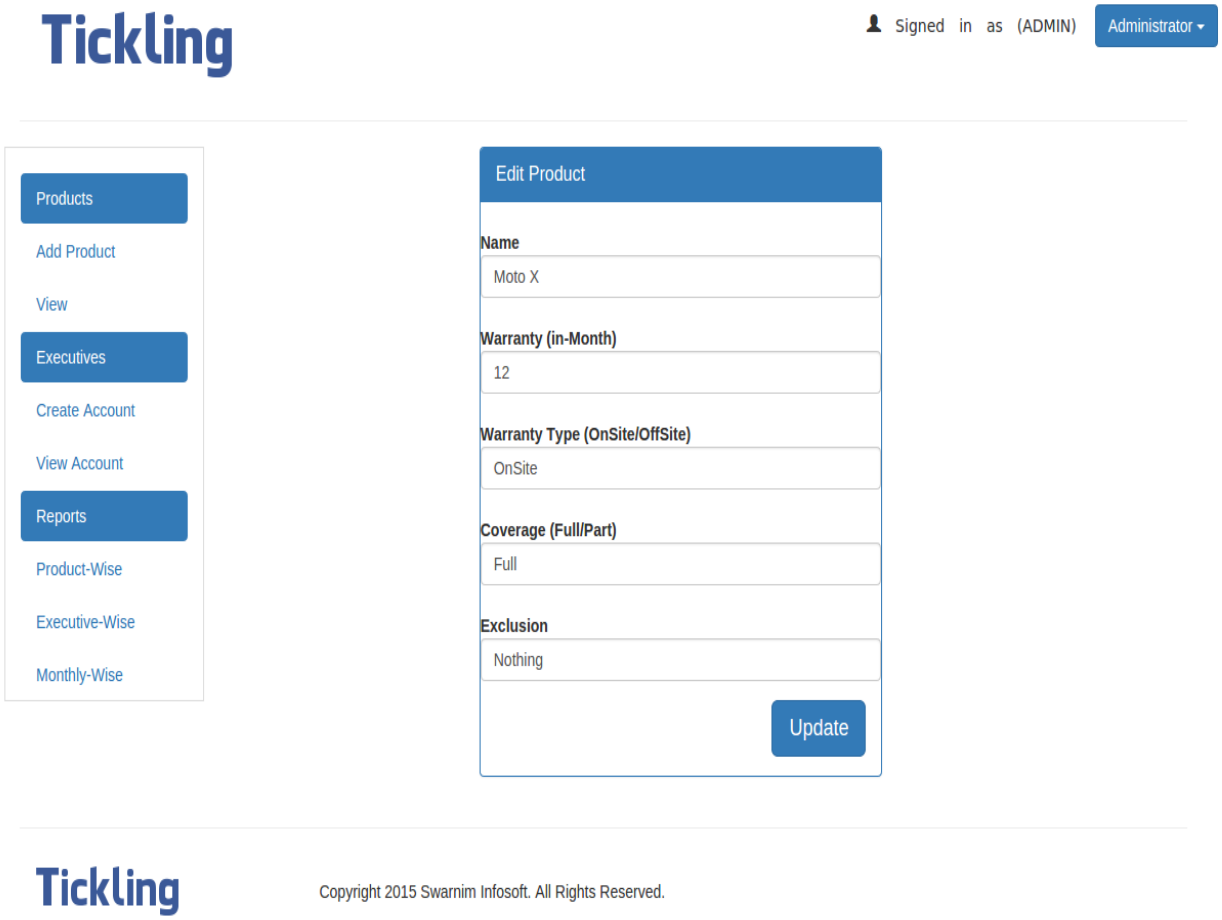


Fig:- Admin can Edit Product Details.

Executive Registration Page:

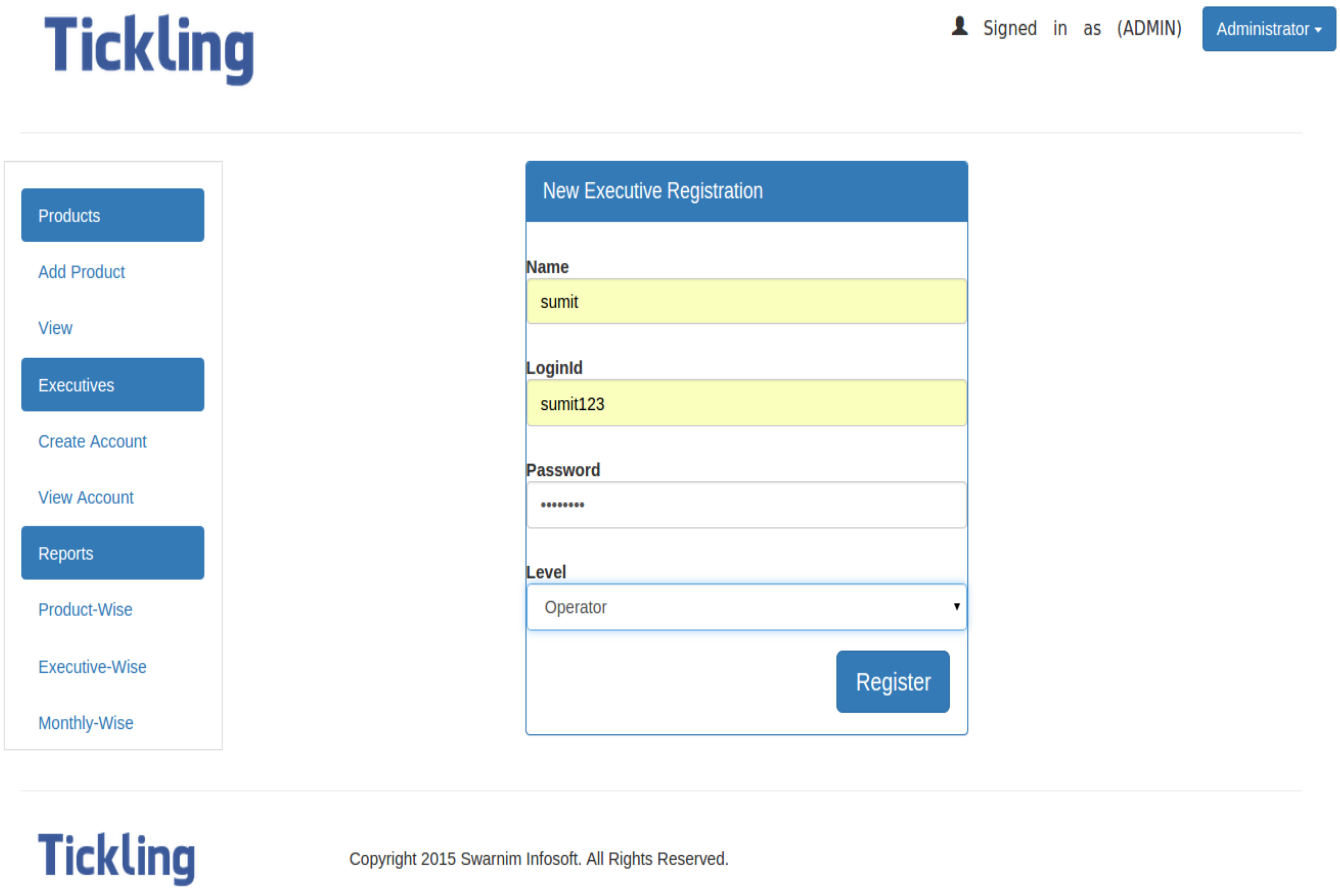


Fig:- Admin can Add Executive For Handling Product Tickets.

Executive List:

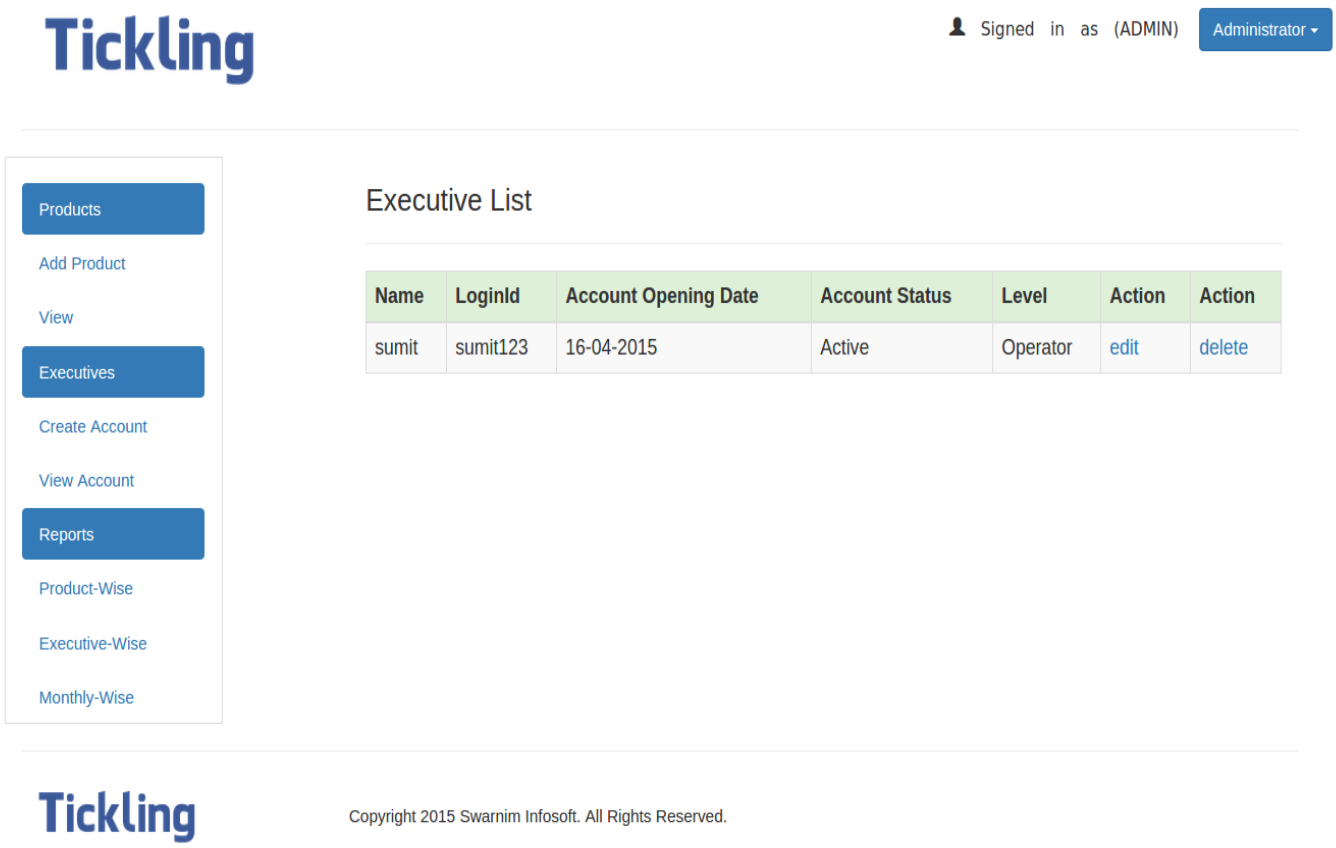


Fig:- Admin can View And Delete Executives.

Edit Executive:

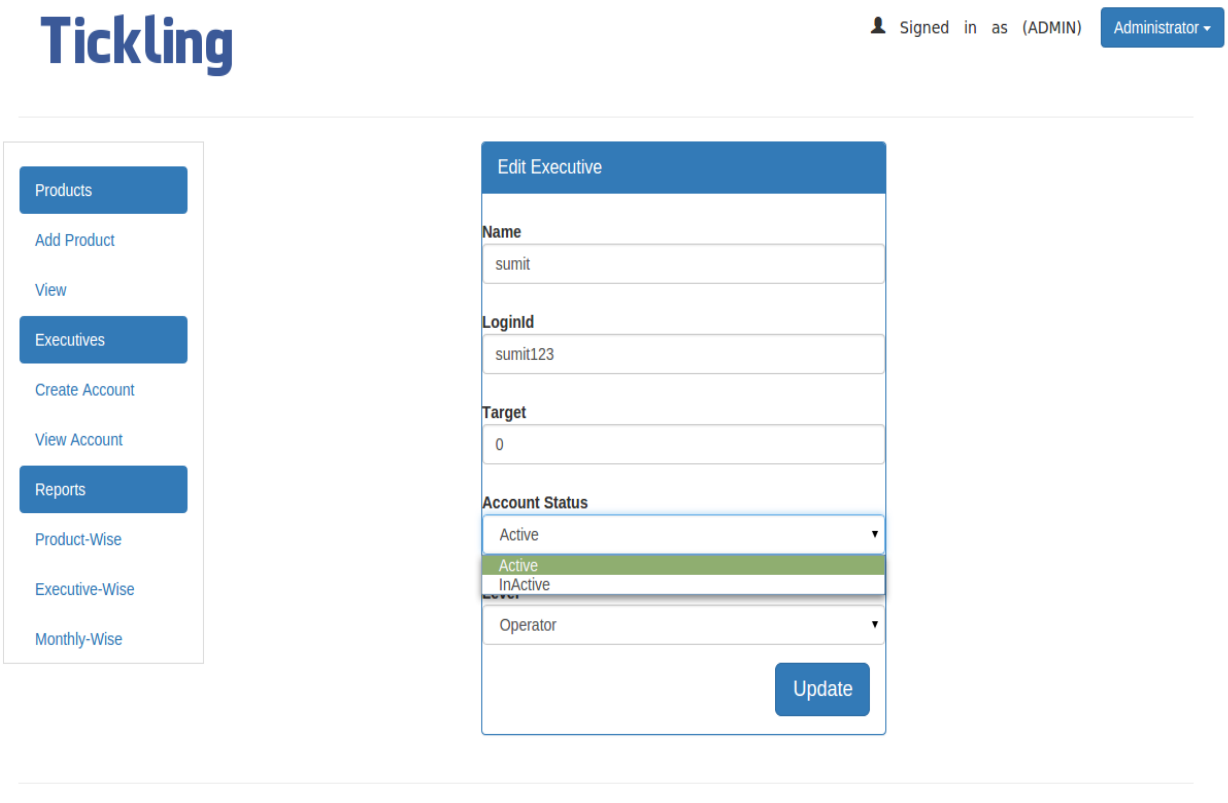


Fig:- Admin can Edit Executive Details.

Profile Page:

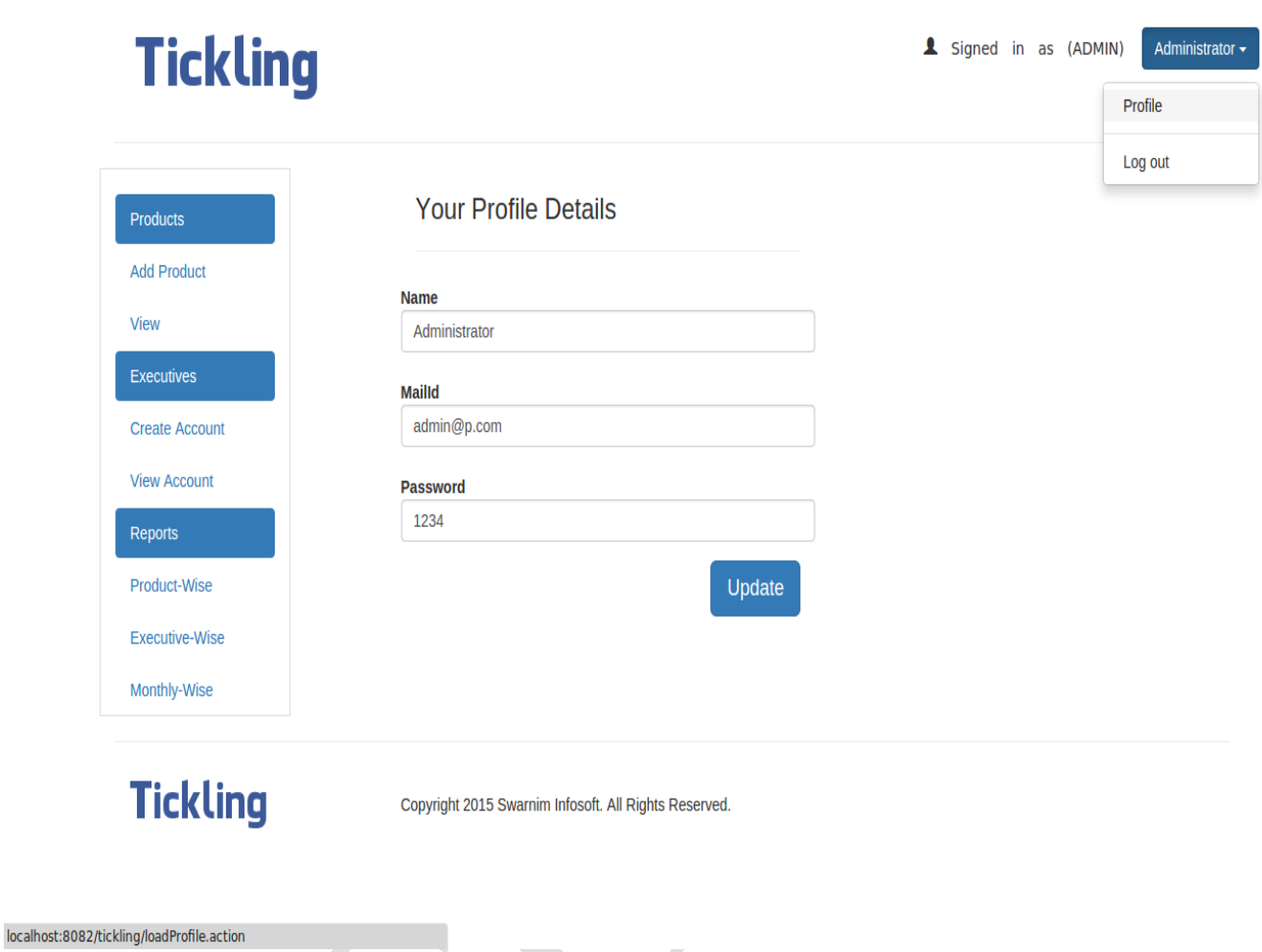


Fig:- Admin and User can Update their Profile.

Welcome Screen for User:

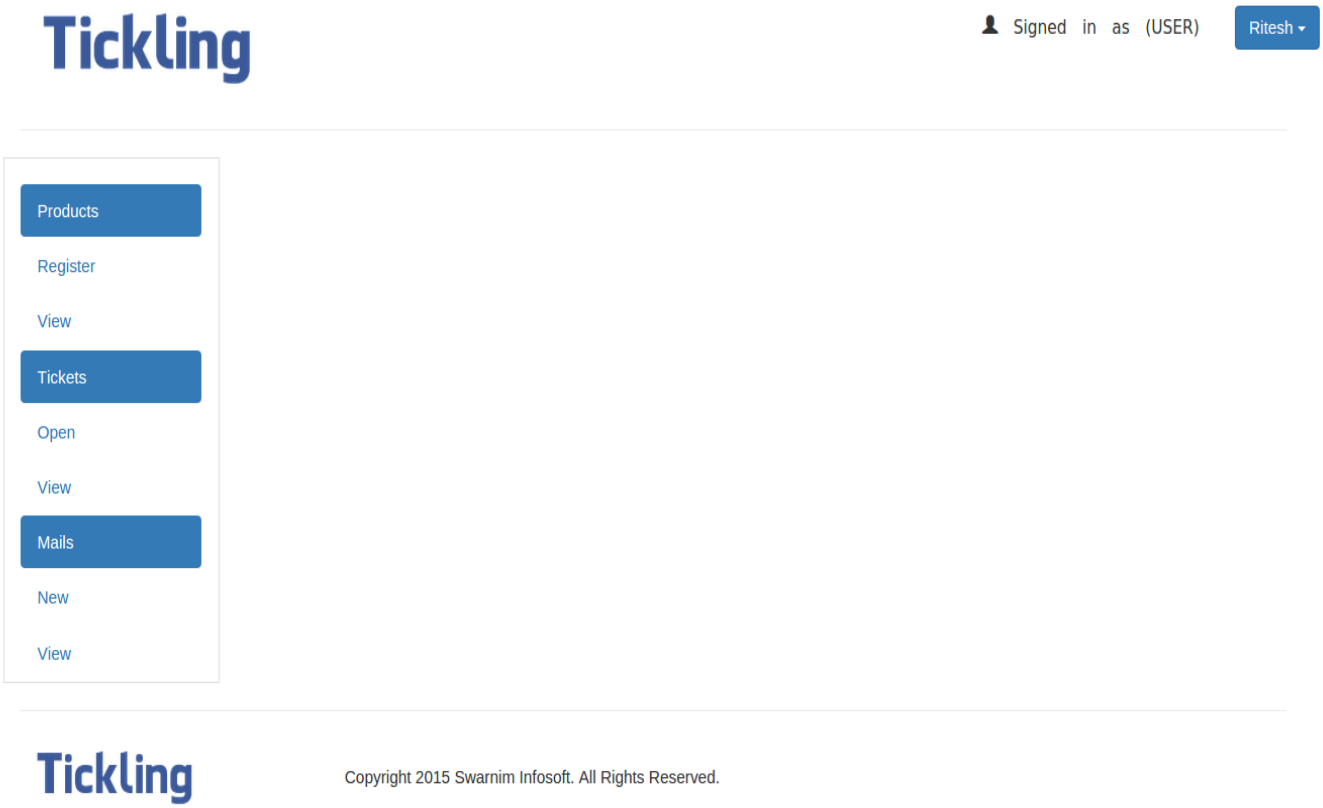


Fig:- User Welcome Screen.

Register Form of Prouct:

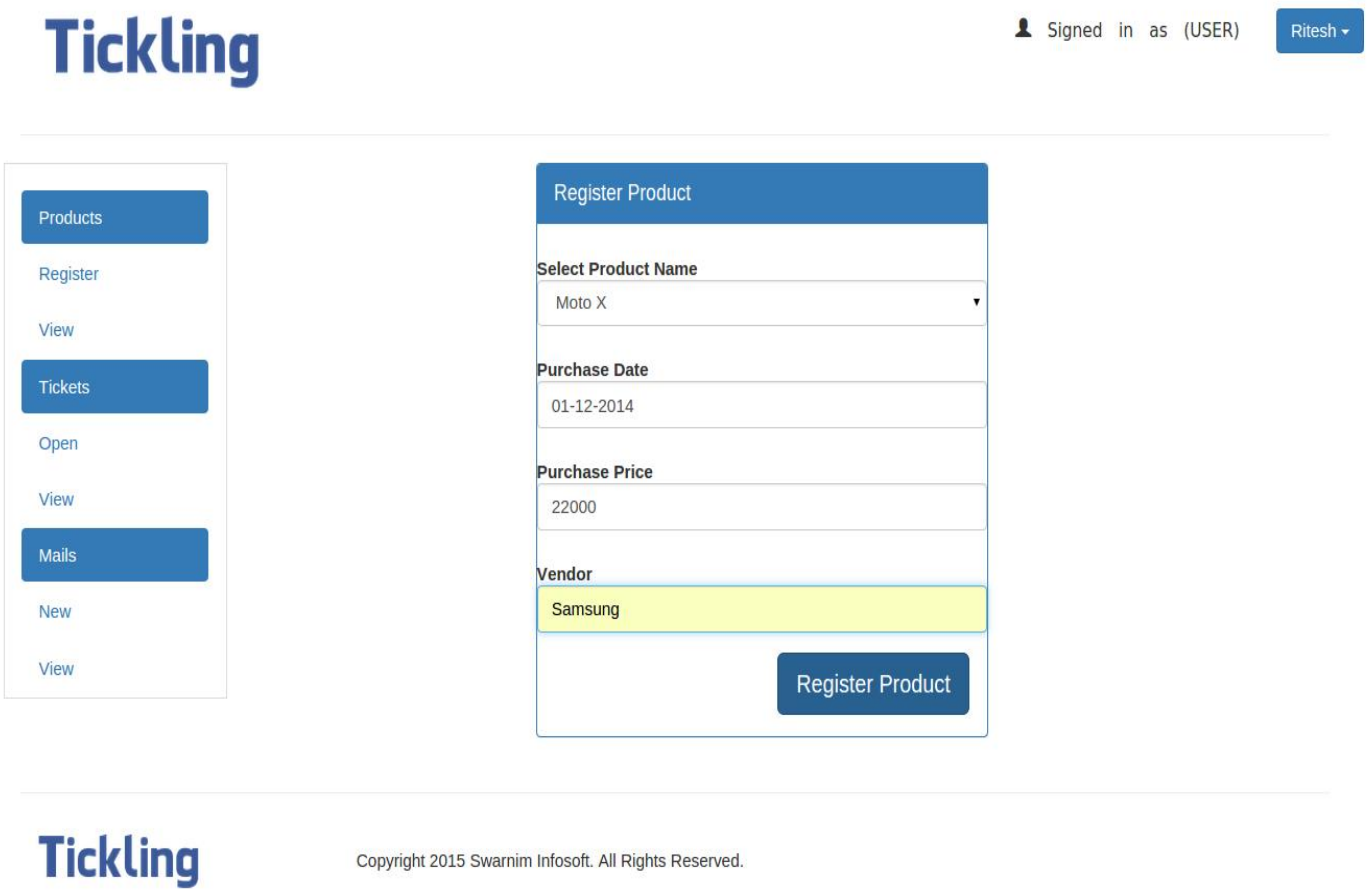


Fig:- Registration form of Product for Warranty Record.

Product List Page:

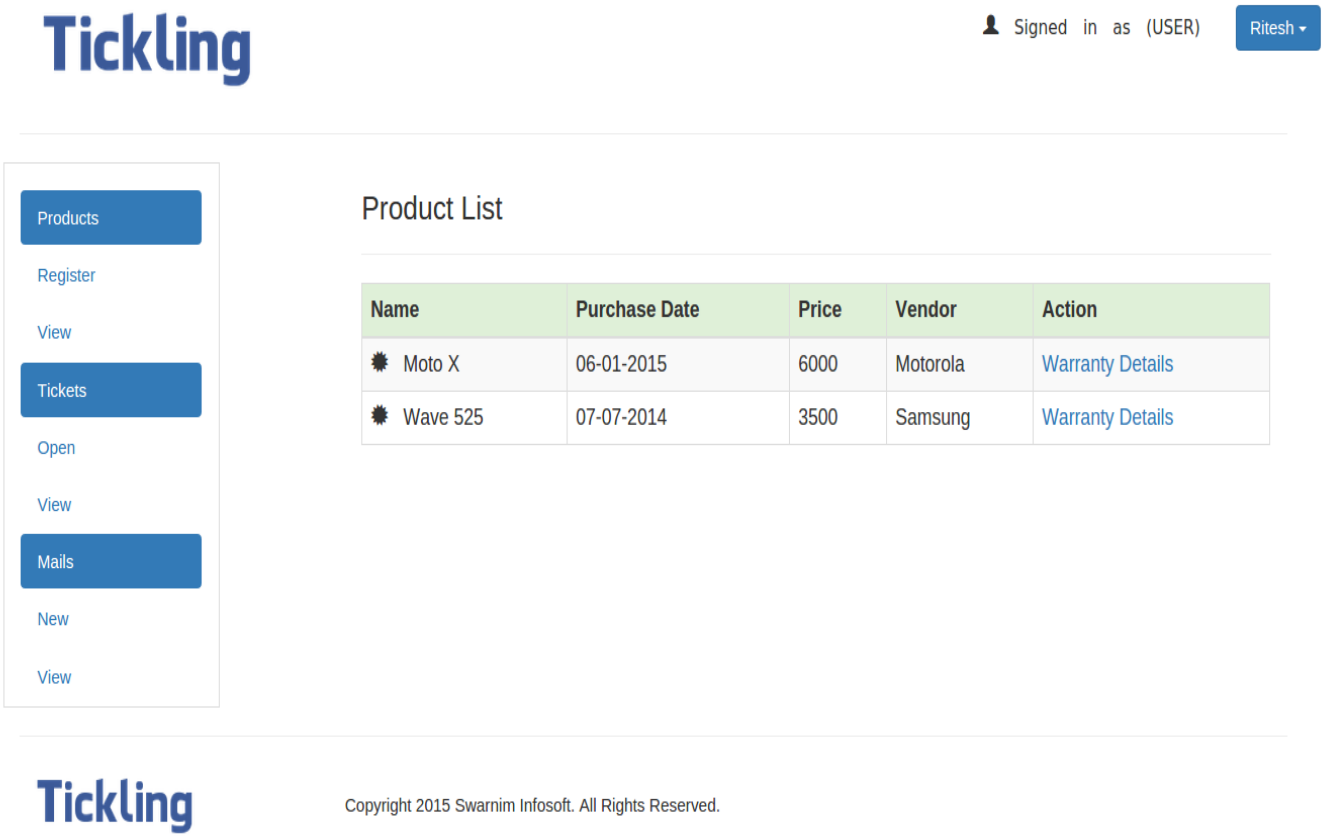


Fig:- List of Registred Product by User.

Warranty Details:

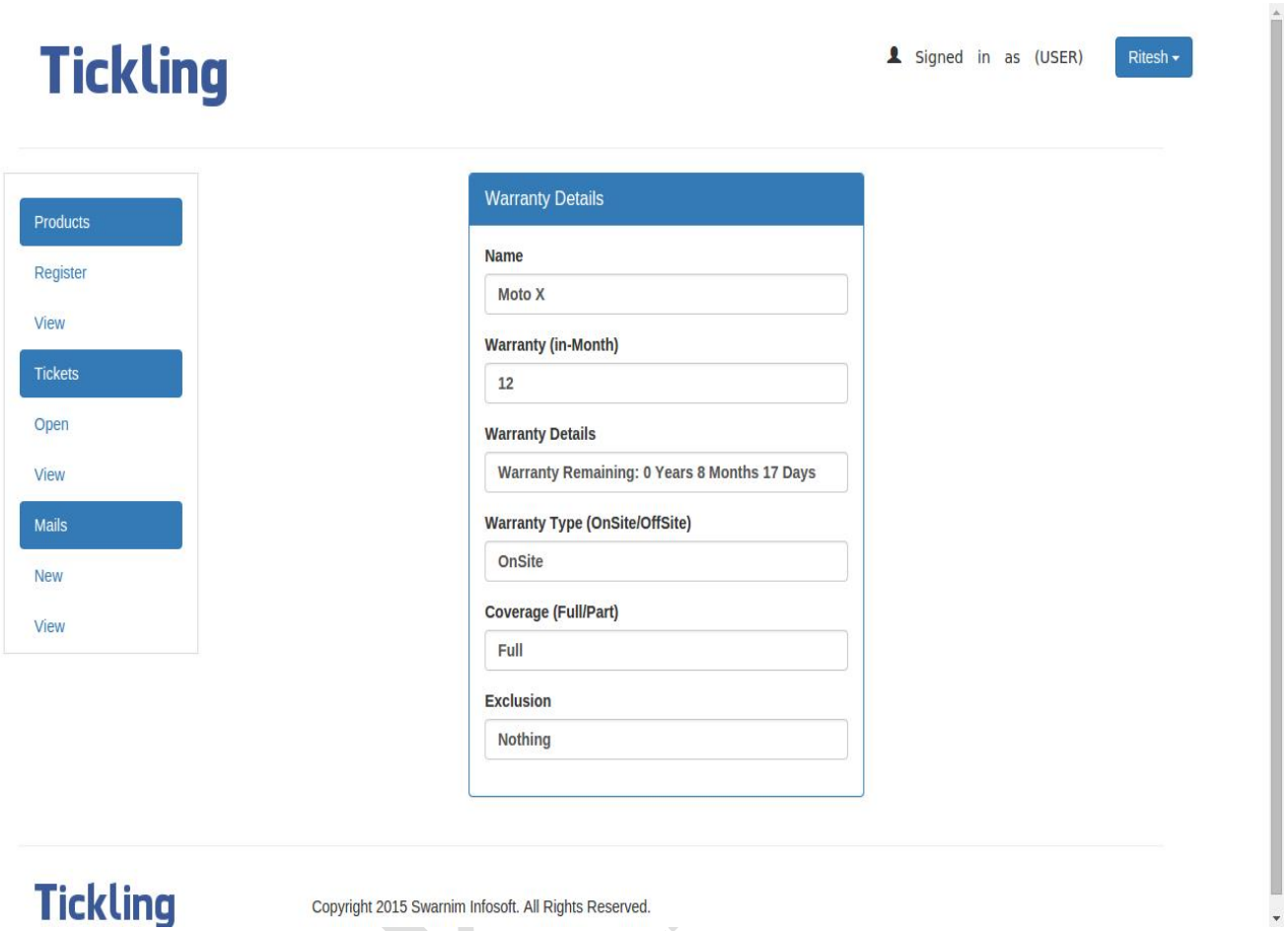


Fig:- Warranty details of a Product.

Ticket Open:

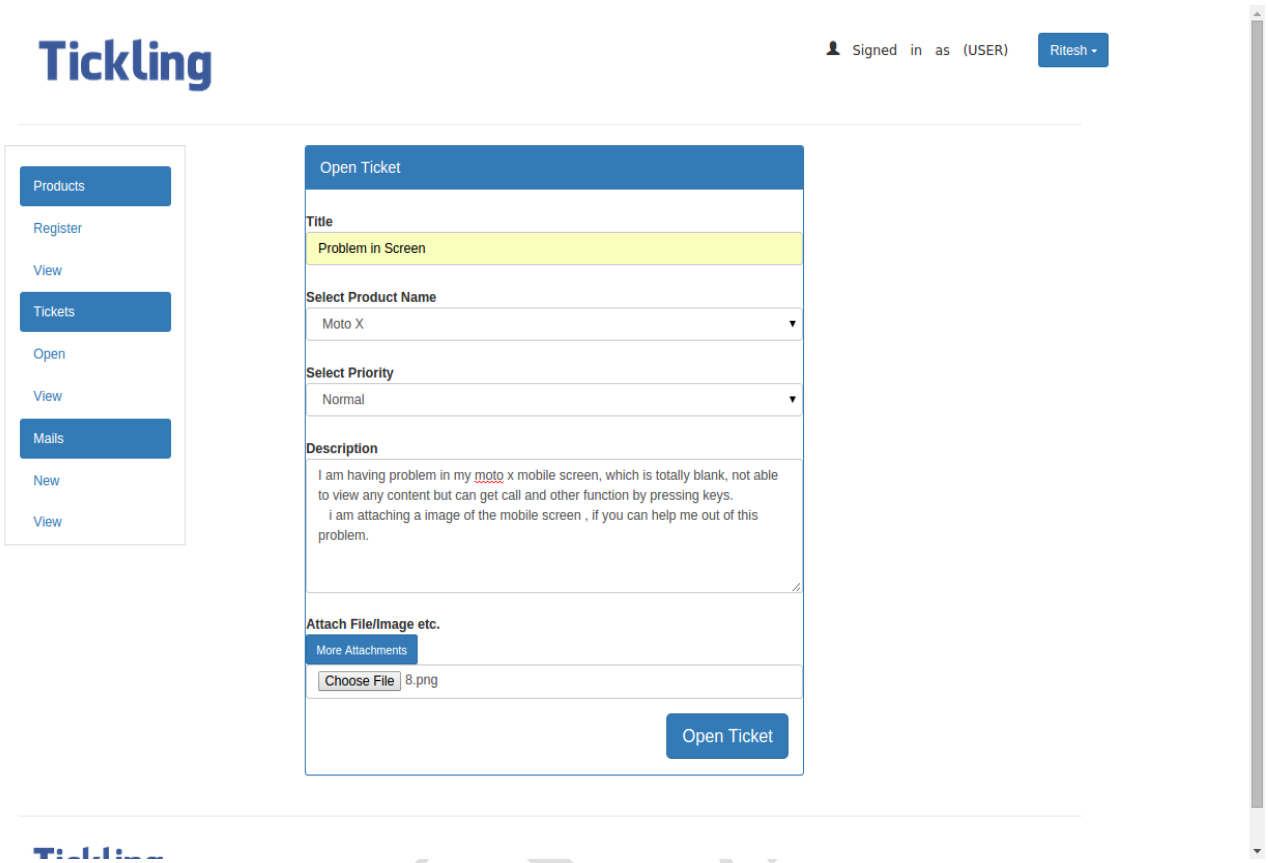


Fig:- Ticket Open Form if any problem occured in Product.

List Of Tickets:

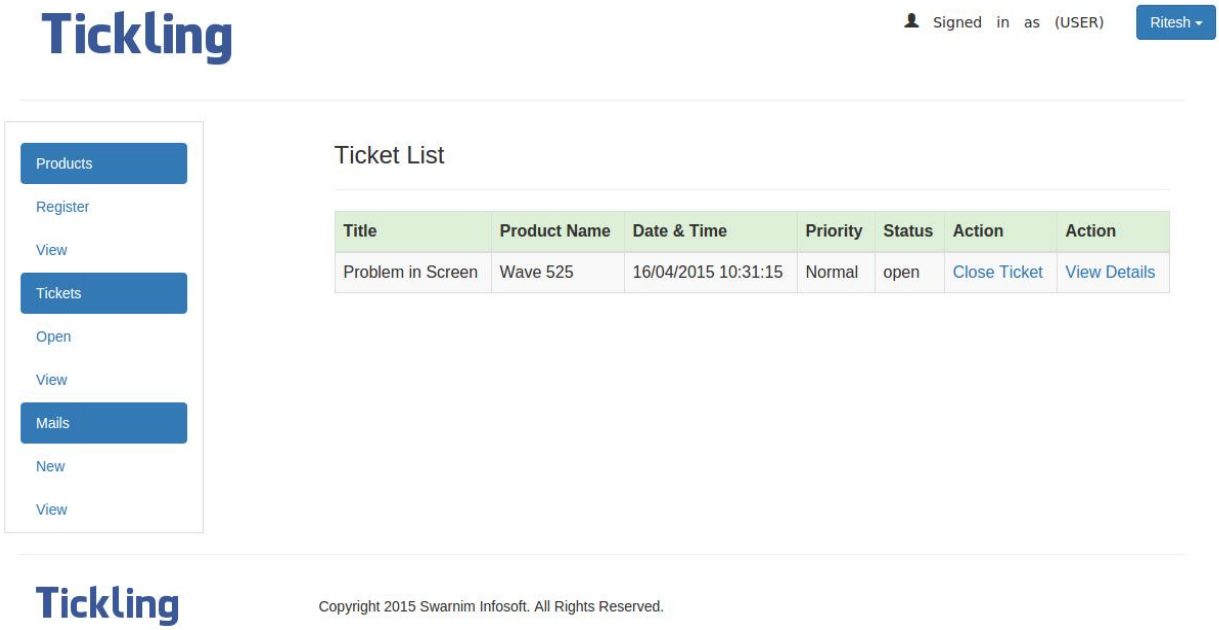


Fig:- View Opened Tickets.

Ticket Details Page:

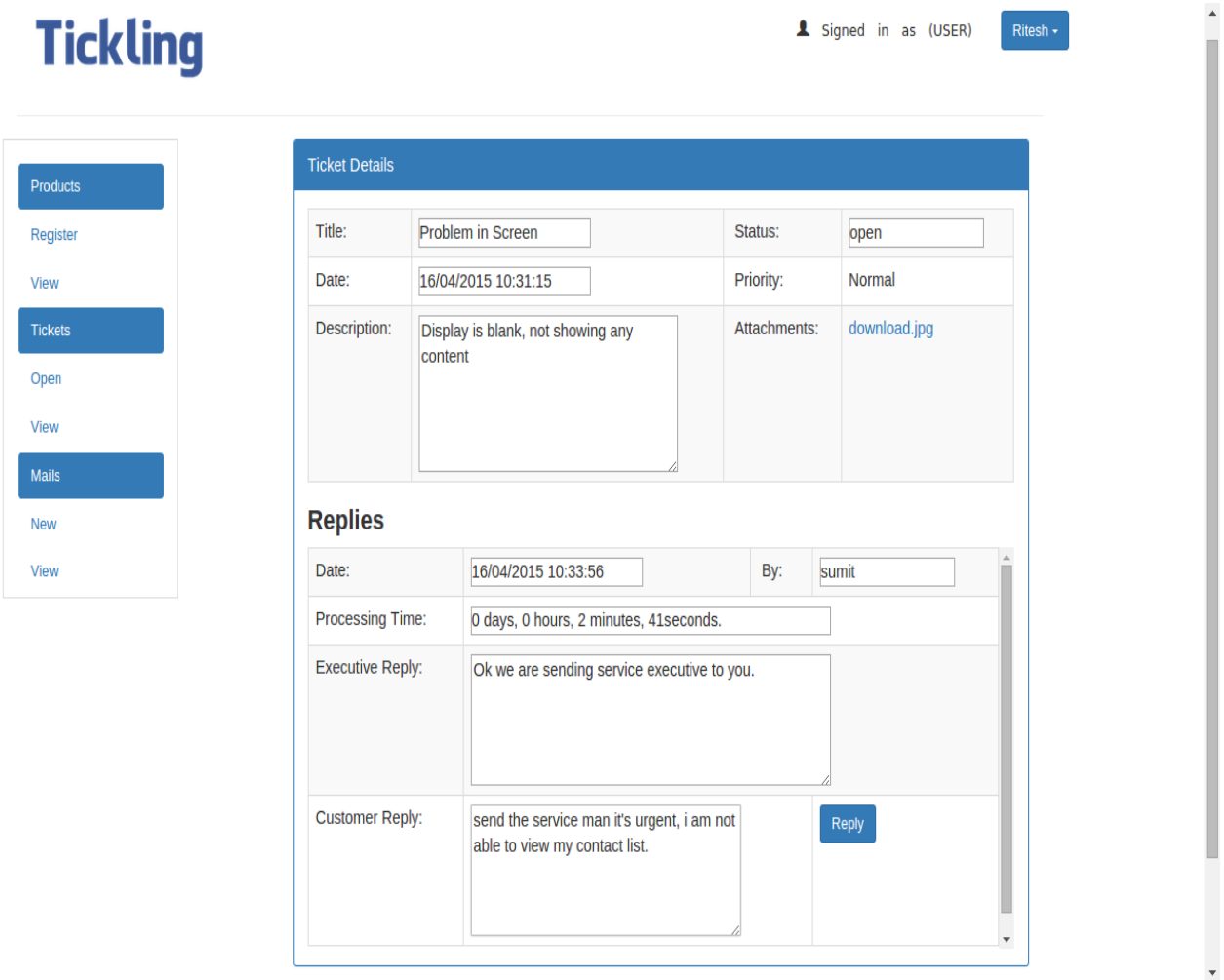


Fig:- View Ticket Details and Replies on it By Executives.

Send Mail to Excutives:

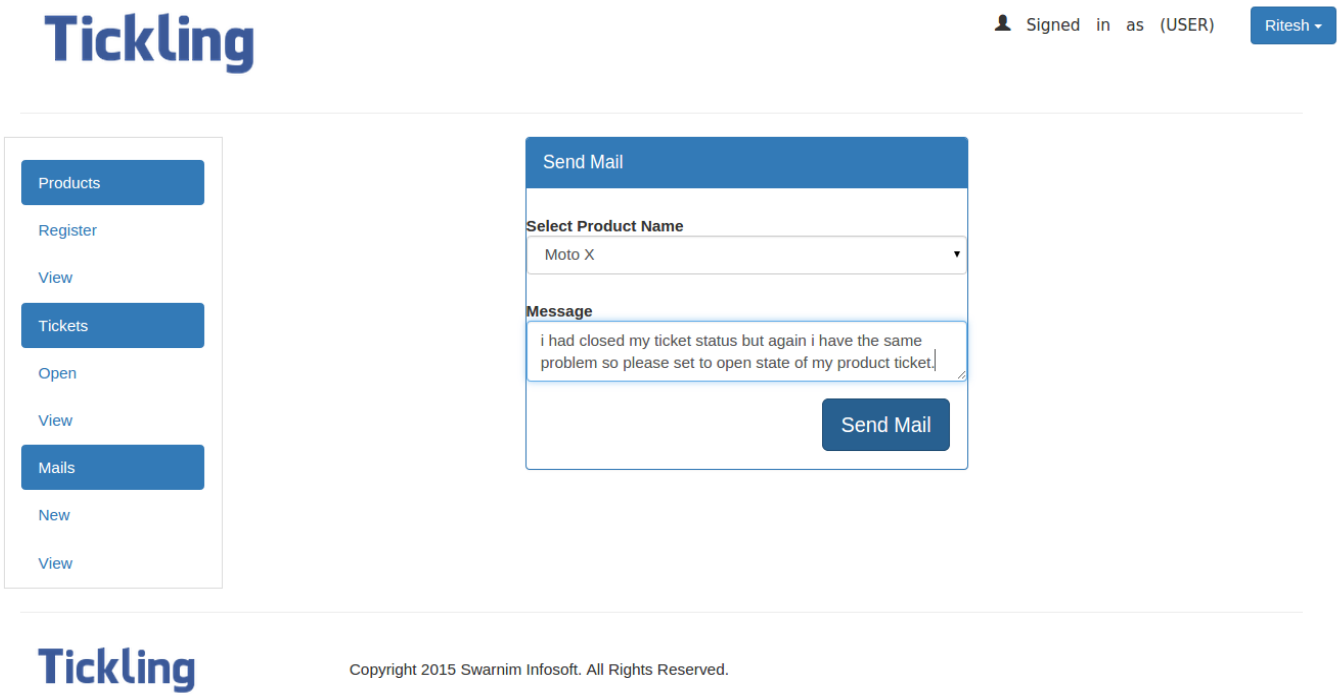


Fig:- Send Mail to executive for Ticket status reset to Open if problem not solved.

Received Mail List:

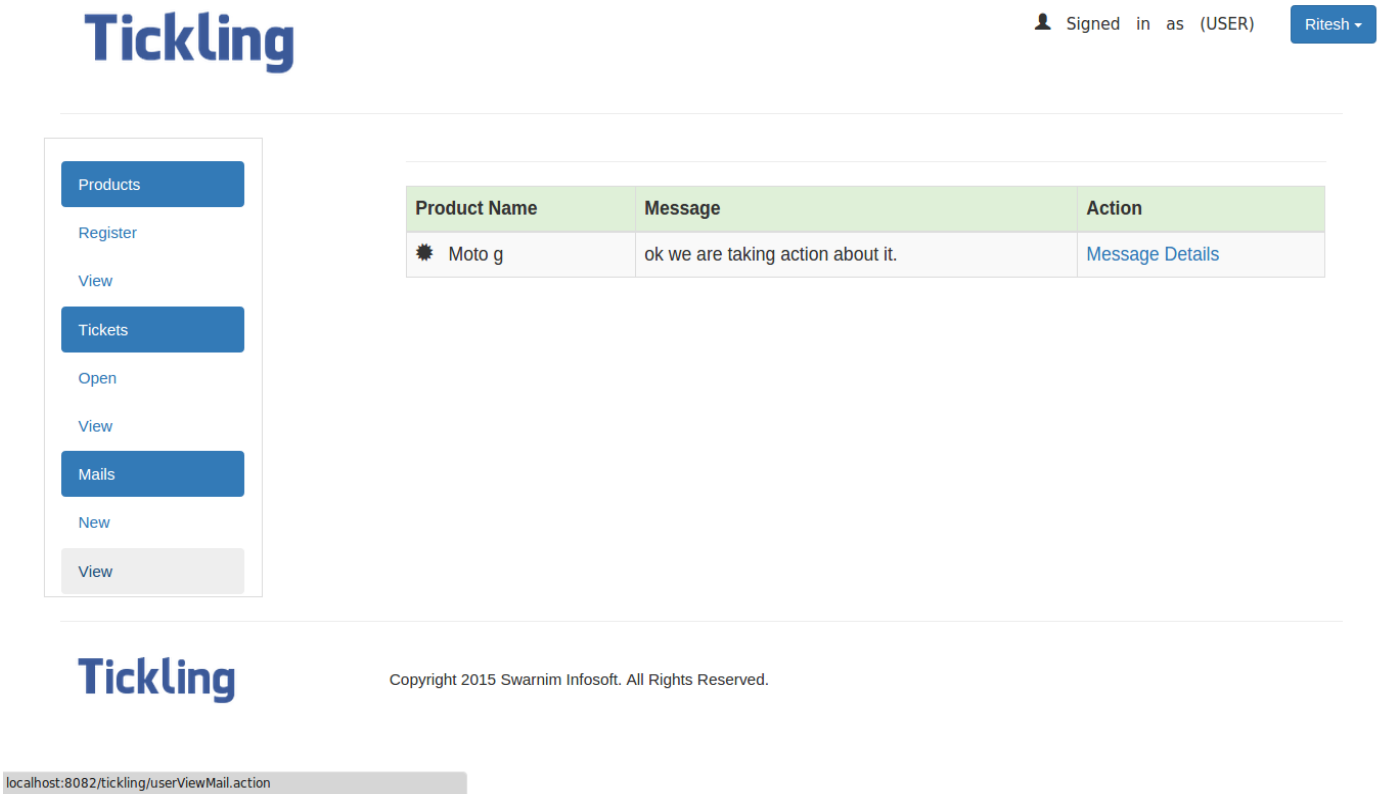


Fig:- View Received mails from Executive.

Details of Received Mail:

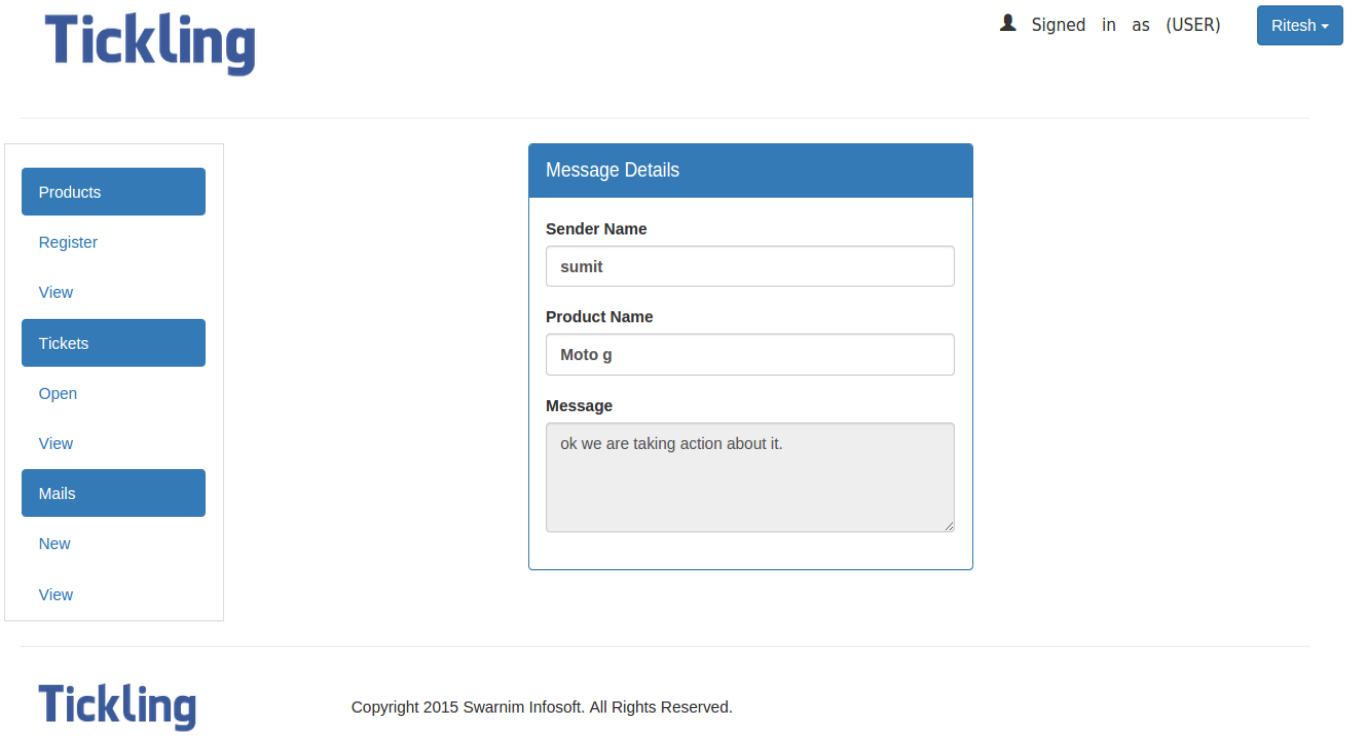


Fig:- View Received mail detail from Executive.

Welcome Screen for Executive:

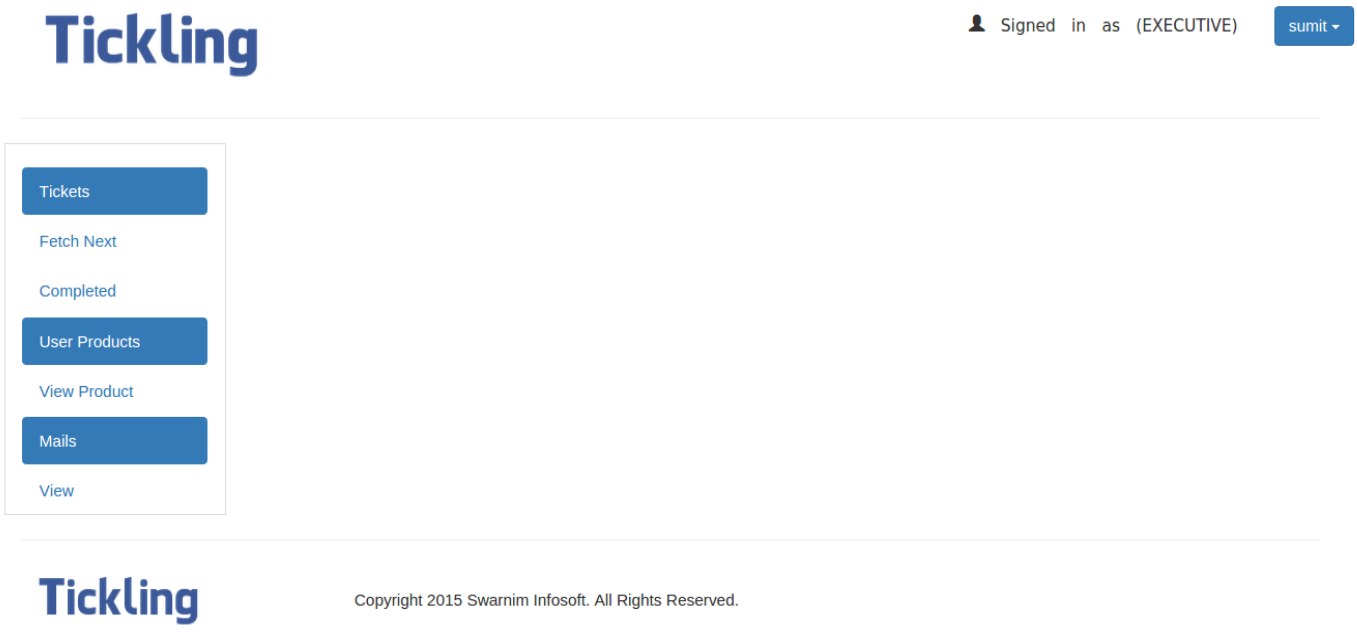


Fig:- Executive Welcome Screen.

Ticket Details:

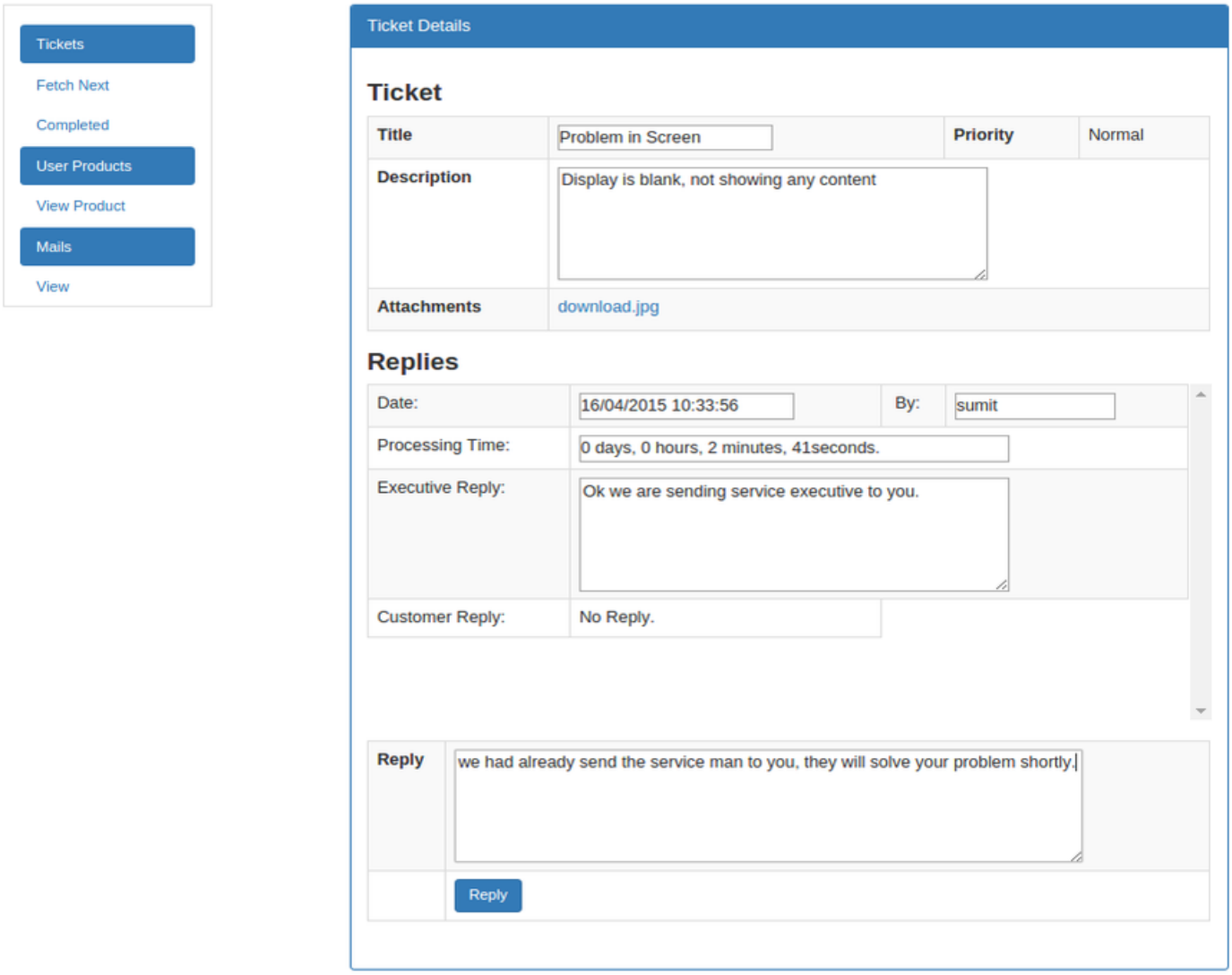


Fig :- Ticket Fetched from Queue.

Ticket List:

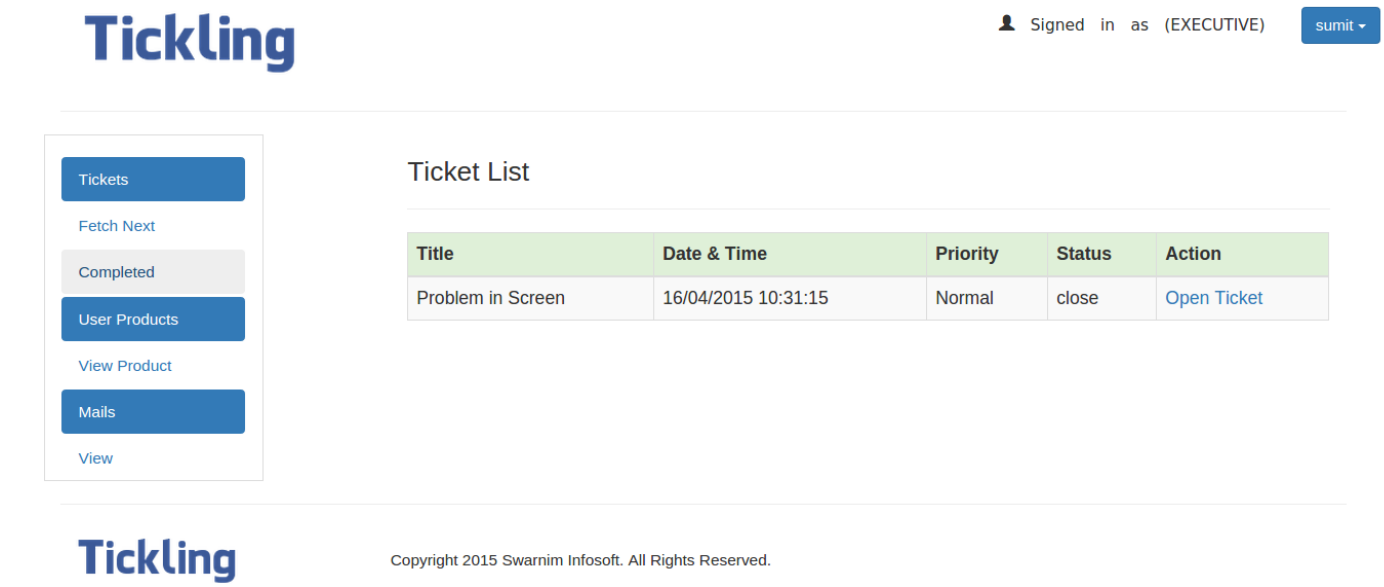


Fig:- View Completed Tickets.

Product List:

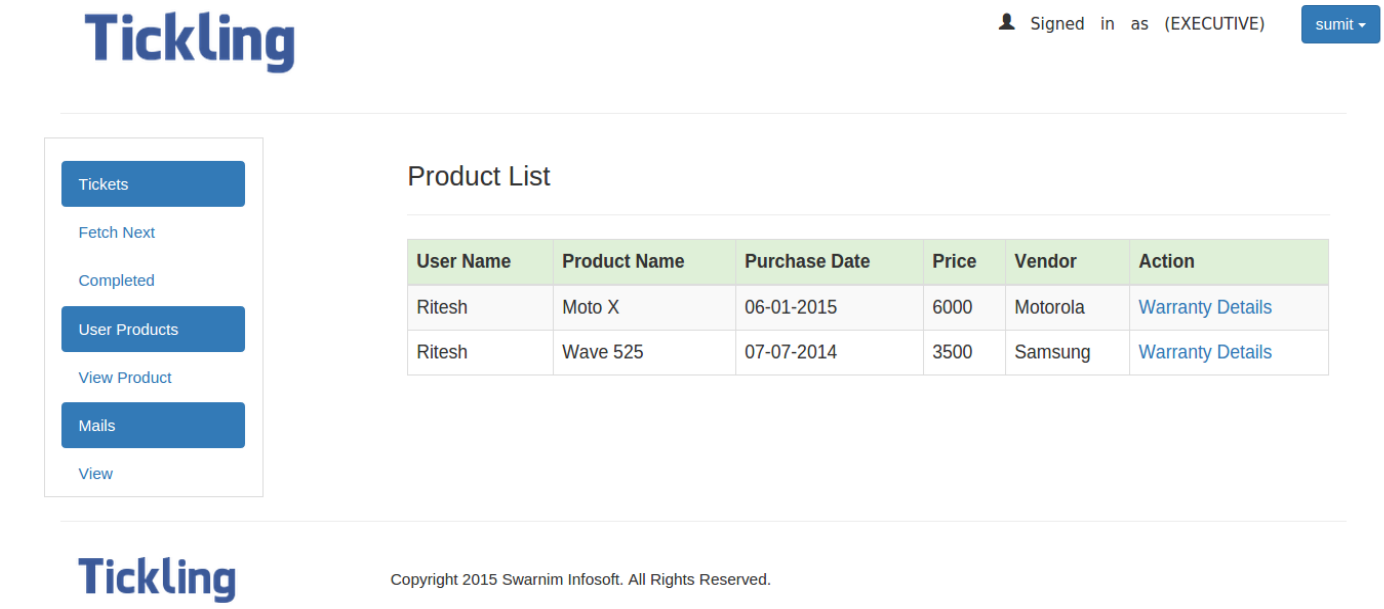


Fig:- View Registered Products.

Warranty Details:

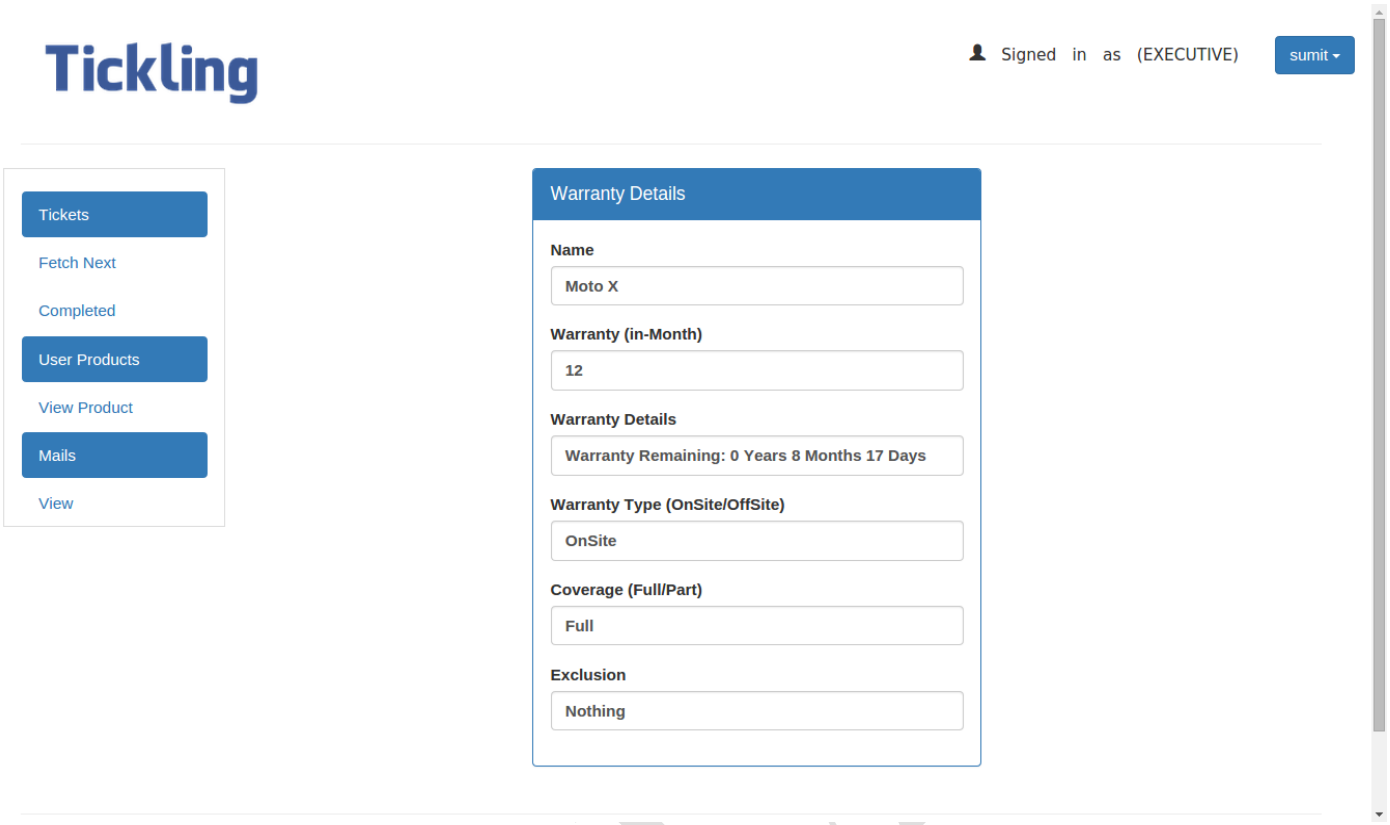


Fig:- View Registered Product Detail.

View Mail from User:

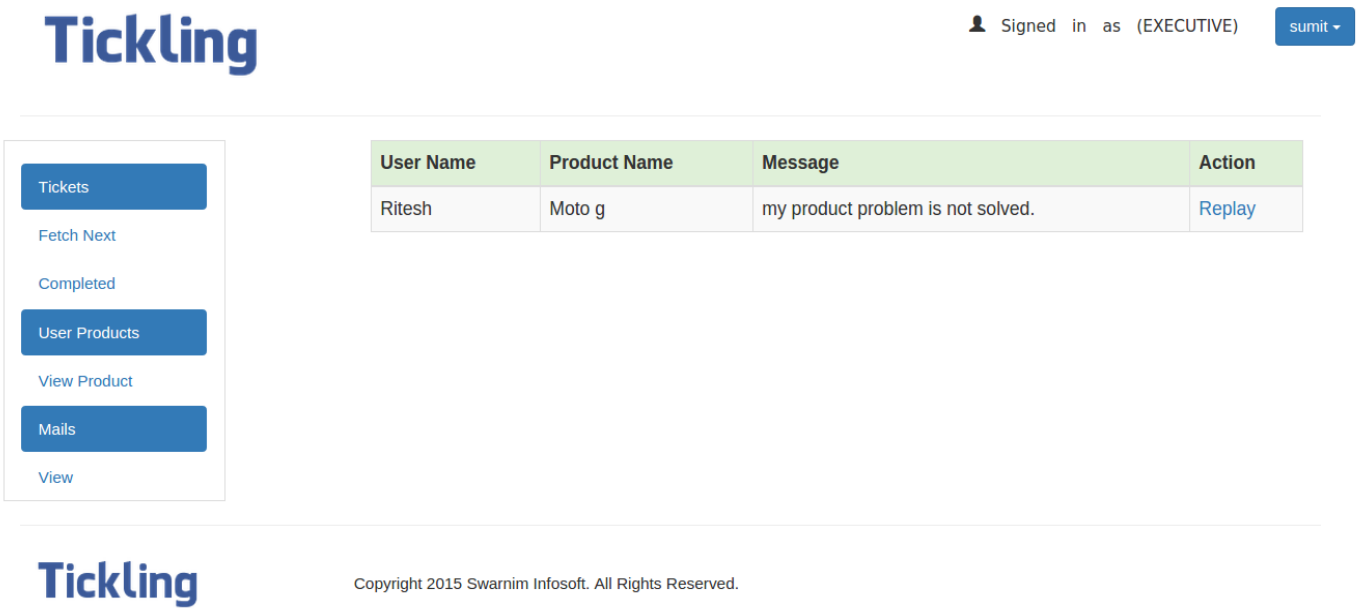


Fig:- View Mail from User.

Reply Pages:

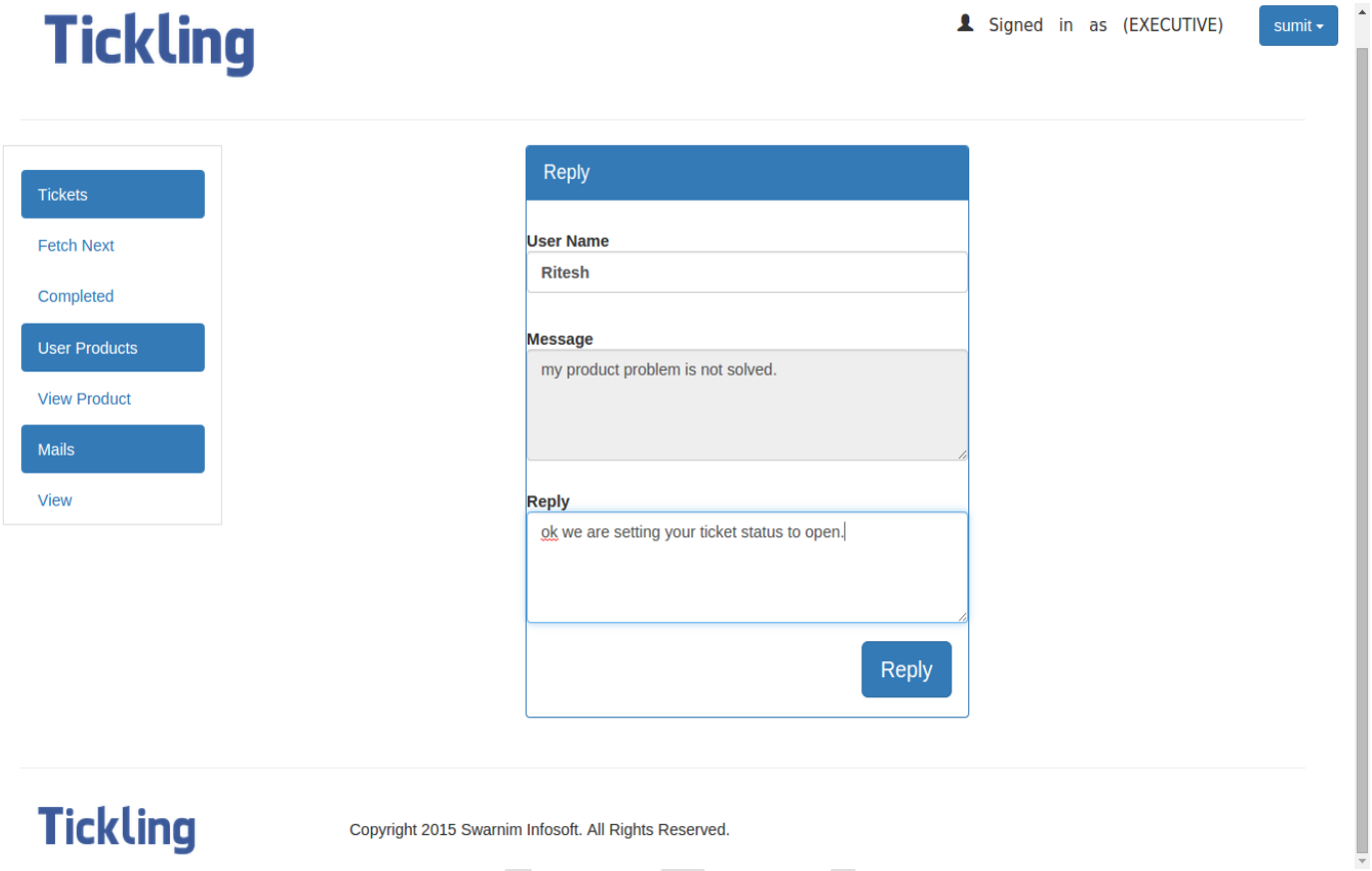


Fig:- View Mail Detail and Reply on it.

Report Product Wise Admin View:

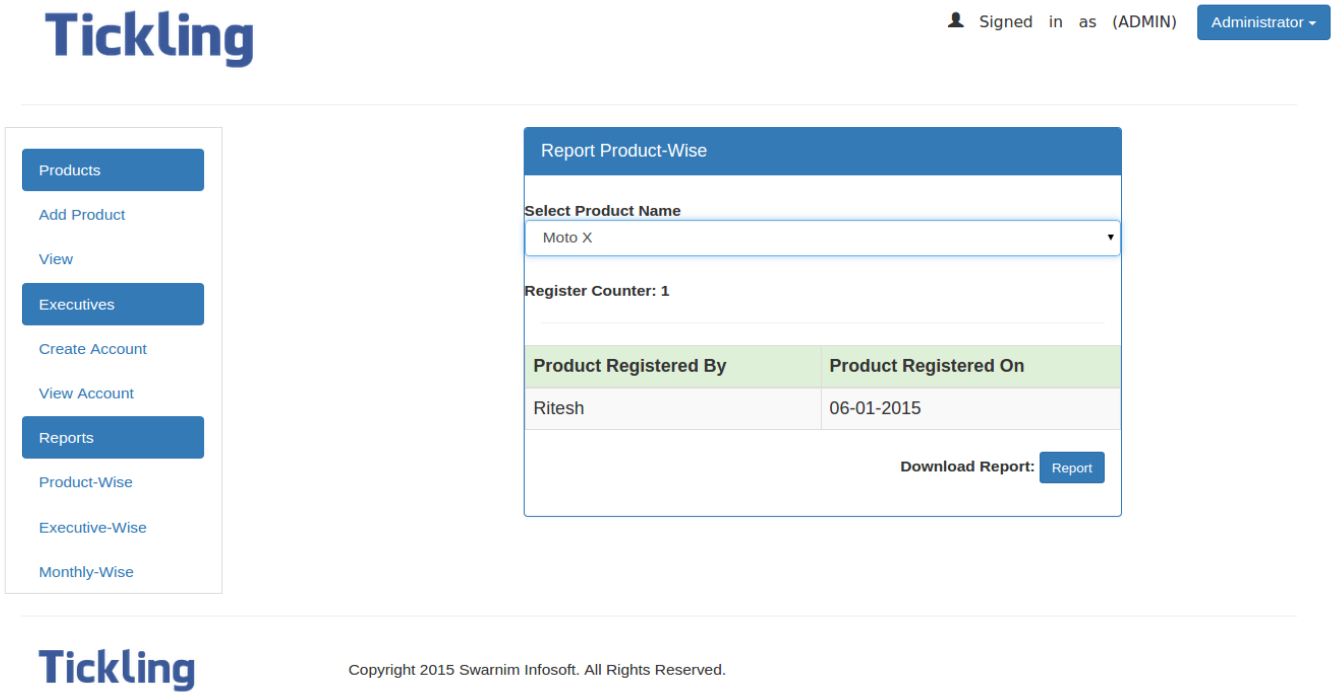


Fig:- Admin view Report Product-Wise

Admin Download Report Product-Wise:

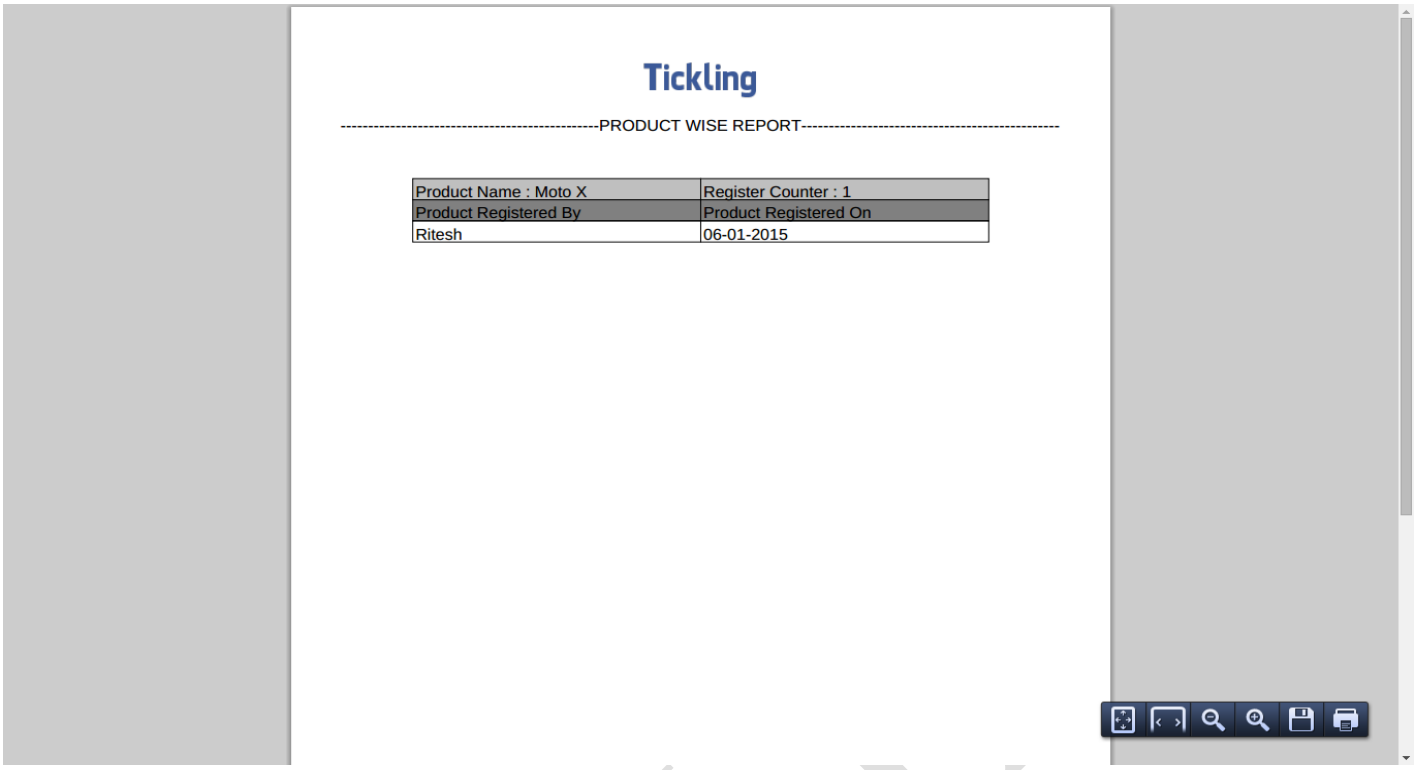


Fig:- Admin Download Report Product-Wise.

View Report Executive Wise:

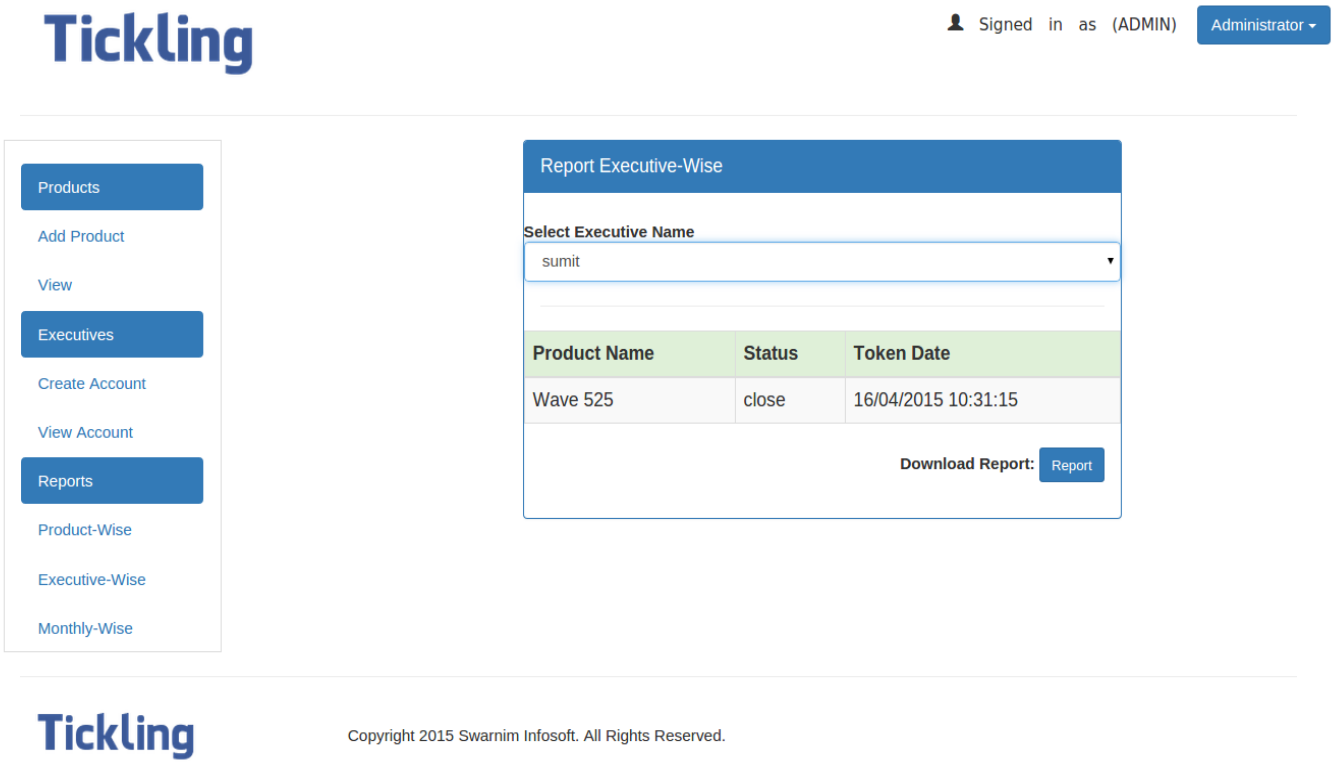


Fig:- Admin view Report Executive-Wise.

Download Report Executive Wise:

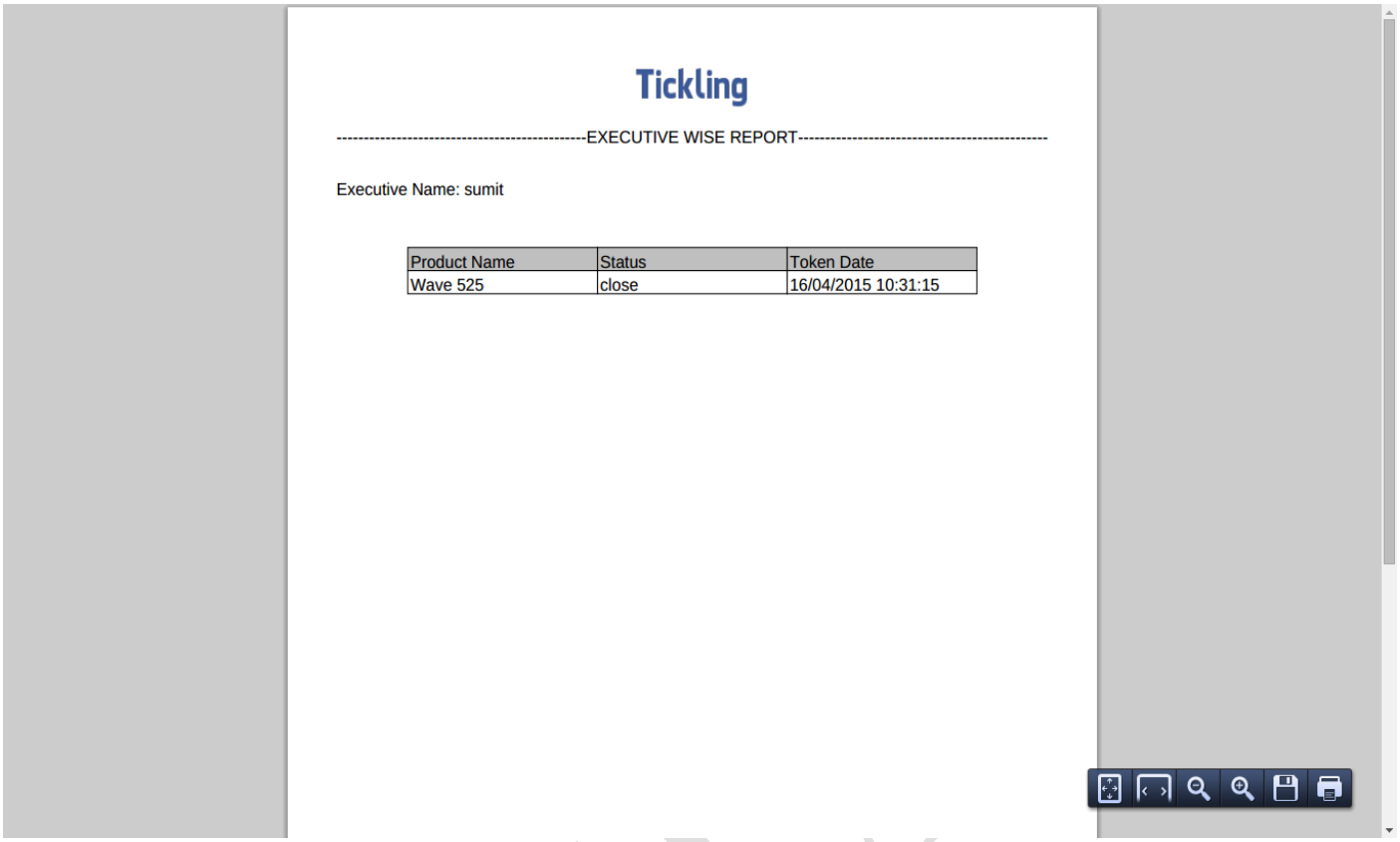


Fig:- Admin Download Report Executive-Wise.

Admin select dates for Mont-Wise Report:

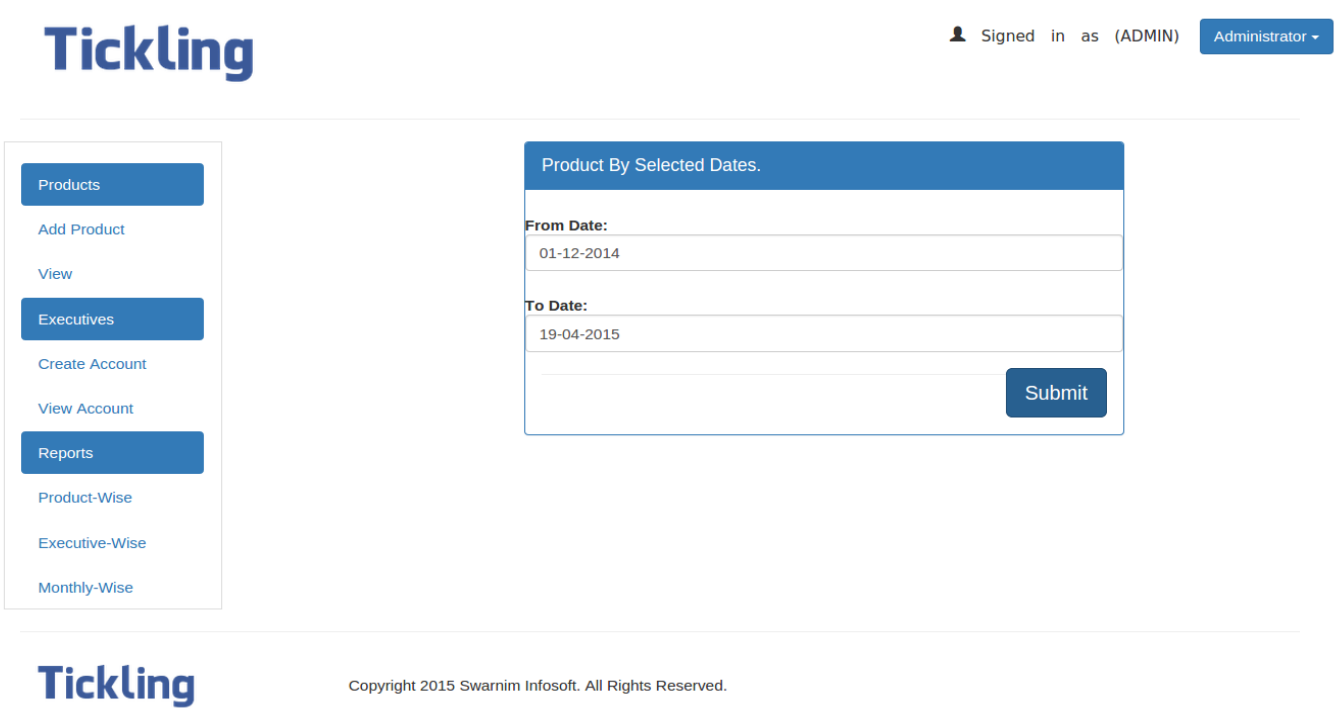


Fig:- Admin select dates for Mont-Wise Report.

View Moto-Wise Report:

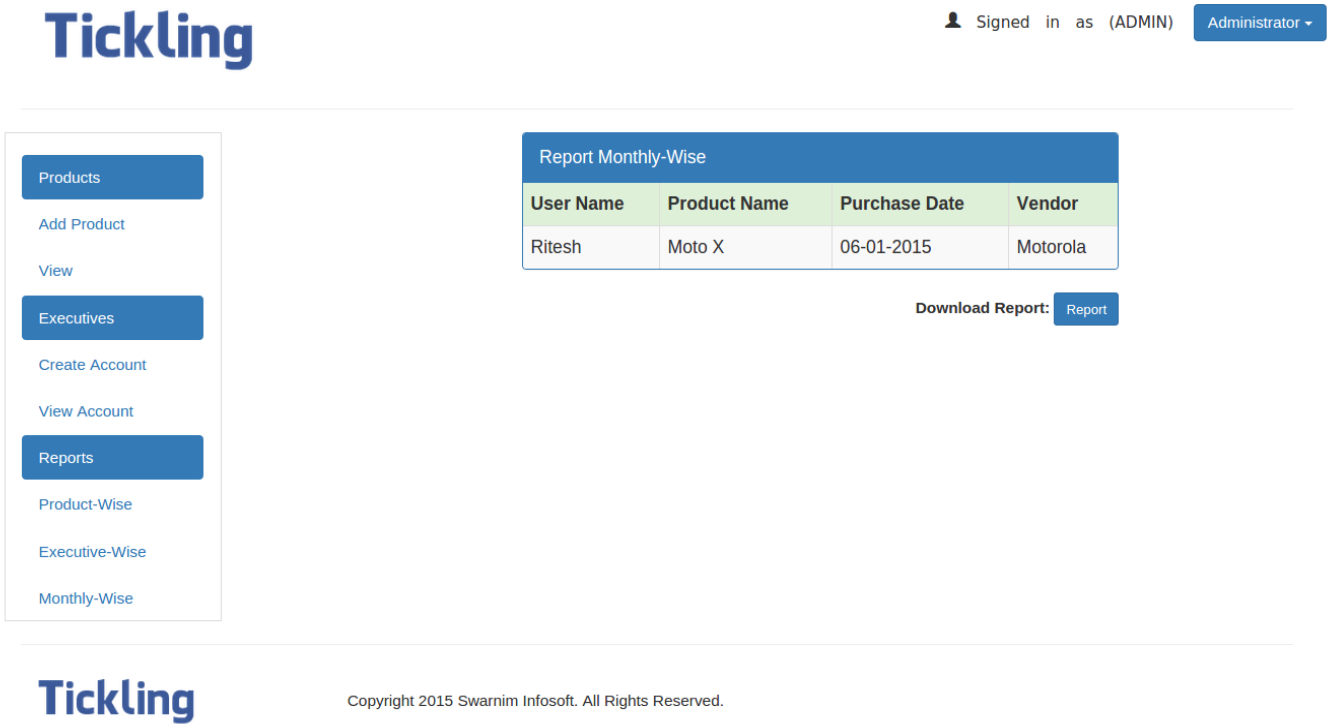


Fig:- Admin view Mont-Wise Report.

Download Month Wise Report:

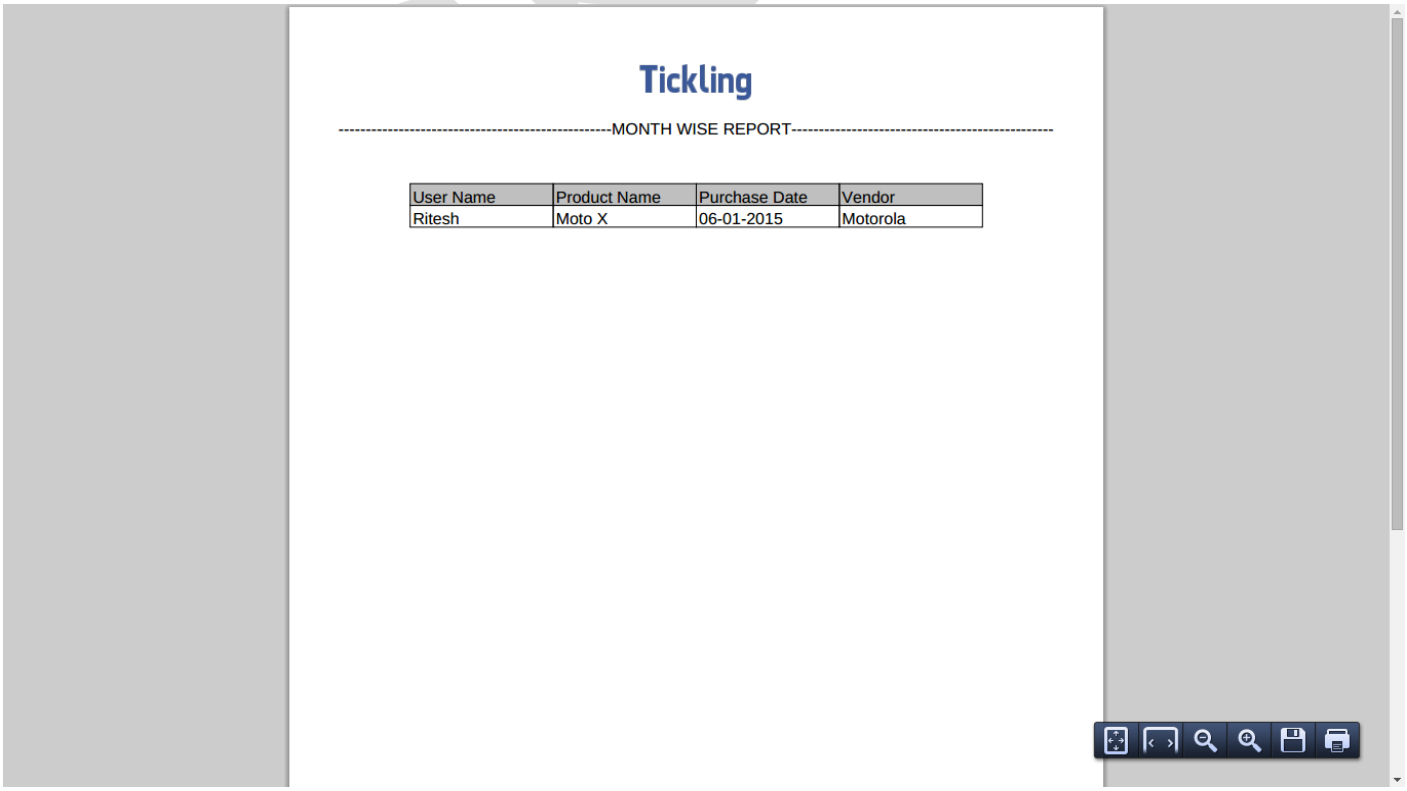
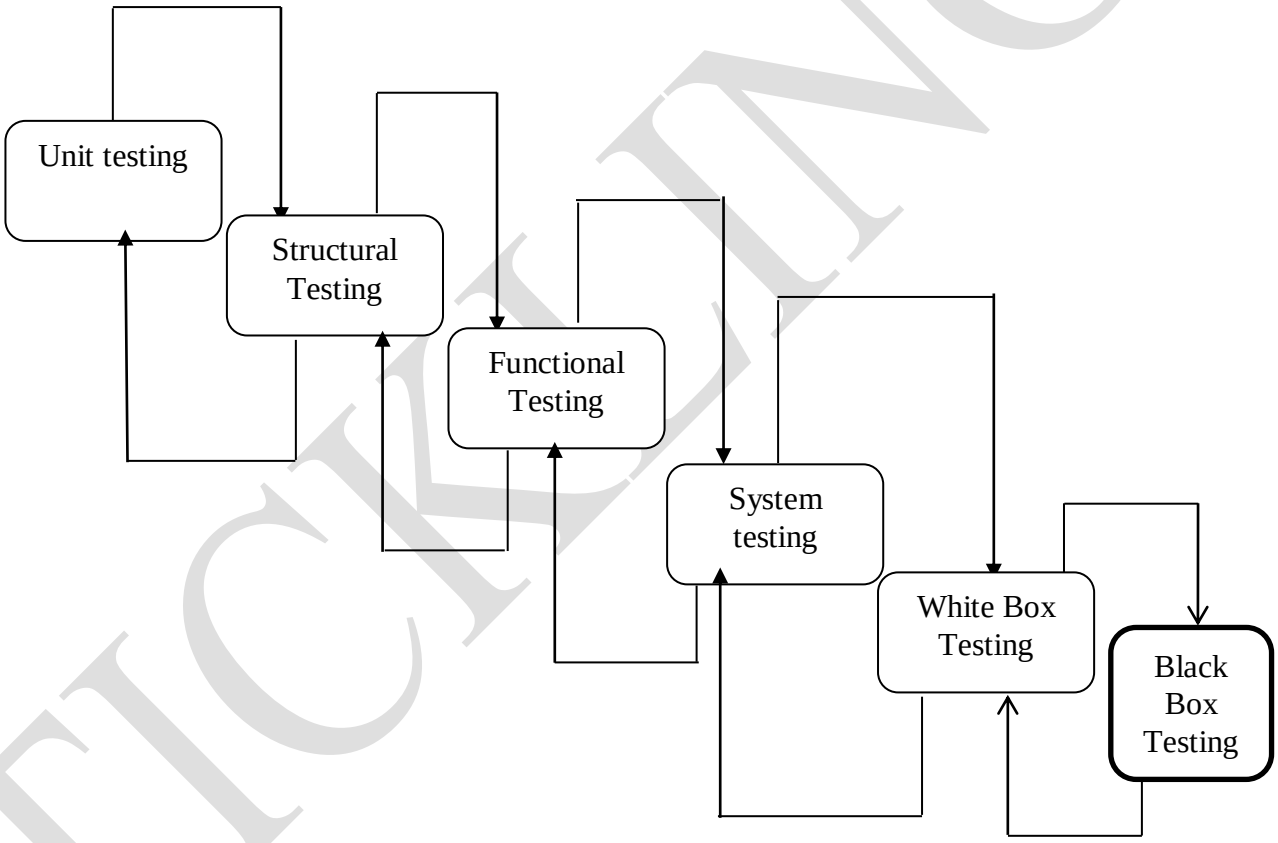


Fig:- Admin Download Mont-Wise Report.

5. Testing

Testing is the set of activities that can be planned in advance and conducted systematically. It is an integral part of program development. It is in this stage, which we check that the program, that has been coded, performs according to the user's requirements. The purpose of doing test is not to demonstrate that there are no errors in the program but to detect any bugs that may still exist. In the testing stage, the main aim is to look for errors that unknowingly have been occurred. It is common misconception that the purpose of testing is to prove that a program is working correctly. This is dangerous myth because it can lead insufficient testing, and program with hidden fault. Because the actual result and expected result may differ in the field of reality and it can be hazardous for a program.



Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of the software. The results of testing are used later on during maintenance also.

Testing is vital to the success of the system. System testing makes a logical assumption that if the parts of the system are correct, the goal will be successfully achieved. In adequate testing or non-testing leads to errors that may not appear until months or even years later (Remember the New York three day power failures due to a misplaced 'Break' statement).

This creates two problems:-

- The time lag between the cause and the appearance of the problem.

- The time interval effect of the system errors on files and the records on the system.

A small error can conceivably explode into a much larger problem. Effective testing early in the process translates directly into long term cost savings from a reduced number of errors. Another reason for system testing is it's utility as a user oriented vehicle before implementation. The best program is worthless if it does not meet the user requirements. Unfortunately, the user's demands are often compromised by efforts to facilitate program or design efficiency in terms of processing time or design efficiency.

Thus in this phase we went to test the code we wrote. We needed to know if the code compiled with the design or not? Whether the code gave the desired outputs on given inputs? Whether it was ready to be installed on the user's computer or some more modifications were needed?

Through the web applications are characteristically different from their software counterparts but the basic approach for testing these web applications is quite similar. These basic steps of testing have been picked from software engineering practices. The following are the steps, we undertook:

❖ Unit Testing:

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications, testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of a module. There are so many modules like Login, HR Department, Interviewer Section, etc. Each module has been tested by giving different sets of inputs. When developing the module as well as finishing the development, the module works without any error. The inputs are validated when accepting them from the user.

❖ System Testing:

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if software meets its requirements.

Here entire 'HRRP' has been tested against requirements of project and it is checked whether all requirements of project have been satisfied or not.

Structural Testing :

Structural testing is concerned with testing the implementation of the program.In structural testing the testers are required to have the knowledge of the internal implementations of the code. Here the testers require knowledge of how the software is implemented, how it works.

The intent of structural testing is not to exercise all the different input or output conditions but to exercise the different programming structures and data structures used in the program.During structural testing the tester is concentrating on how the software does it. For example, a structural technique wants to know how loops in the software are working. Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software.

Structural testing can be used at all levels of testing. Developers use structural testing in component testing and component integration testing, especially where there is good tool support for code coverage. Structural testing is also used in system and acceptance testing, but the structures are different. For example, the coverage of menu options or major business transactions could be the structural element in system or acceptance testing.

❖ Functional Testing :

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes *what* the system does.

Functional testing differs from system testing in that functional testing verifies a program by checking it against design documents or specifications, while system testing validates a program by checking it against the published user or system requirements.

Functional testing typically involves following steps:

1. The identification of functions that the software is expected to perform
2. Creation of input data and determination of output based on the function's specifications
3. The execution of the test case and comparison of actual and expected outputs.

❖ WHITE BOX TESTING:

This is a unit testing method, where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors.

I tested step wise every piece of code, taking care that every statement in the code is executed at least once. The white box testing is also called Glass Box Testing.

I have generated a list of test cases, sample data, which is used to check all possible combinations of execution paths through the code at every module level.

White-box test focuses on the program control structure. Test cases are derived to ensure that all statement in the program control structure. Test cases are derived to ensure that all statement in the program control structure. Test cases are derived to ensure that all statement in the program control structure.

❖ BLACK BOX TESTING:

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details at statement level. Here the module will be treated as a block that will take some input and generate output. Output for a given set of input combinations are forwarded to other modules.

Black-box test are designed to uncover errors functional requirement without regard to the internal workings of a program. Black-box testing techniques focus on the information domain of the software, deriving test cases by partitioning the input and output domain of a program in manner that provides through test coverage.

6. Implementation and deployment

Implementation means execution of the application. It is installing the software to the destination and make it to work there. Special care has to be taken for the implementation of the software. Implementation is the part of the Software Development Life Cycle or simply software development process where software engineers actually program the code.

After the system is tested completely, it is delivered to the onsite team. The onsite team implements the tested application in the client environment. It involves all the activities needed to make the software operational for its users.

The main activities in the implementation stage are planning and defining the process for rollout, to deploy the new application, train users on the new system after the rollout has been implemented, and communicate the details of deployment to relevant people. After the system has been coded and tested the next aim and phase was to successfully implement it at the organization. Special care has to be taken for implementation for the software. To implement the project "Bike Showroom Management System" firstly it only requires software's installed on system mentions in software requirement specificationImplementation means install the software to the destination and make it to work there. Implementation is an ongoing process and can be achieved by one of the following methods:

❖ End User Training:

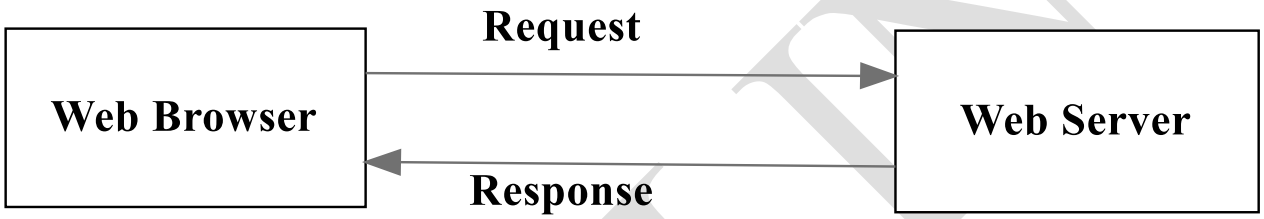
The aim of End User Training is to make users of our business software slutions familiar with the processes surrounding their role. Companies of all sizes spend a significant amount of their IT budgets on software. New desktop operating systems can enhance security and run more sophisticated applications, and those new applications can automate tasks previously done manually or provide easier and faster accomplishment of tasks previously performed using older software, thus enhancing productivity. But we won't see the bottom line benefits of these upgrades unless the end-users of the software can successfully make the transition.

❖ End User Education :

For thesoftware to deliver on its value, end users must develop the skills and knowledge on how to use it effectively. End user adoption is critical to a successful software implementation and its on-going use is only possible with the effective education of the end user.

The website will be deployed to a web server using tomcat, for deploying this create a war file which comprises of various css, java script files, java classes and interfaces, jsp and html pages etc. which are being used in development of the project. Deploy it into apache tomcat webapps folder and create database in the MySql server with the name specified with the same user name which is being used in the website. Notably MySql will be also on the same server for the back end back up. The deployment will provide a URL which is used by the normal user to avail the services by typing the URL in web address in the web browser itself.

By any web browser like Google chrome, Mozilla Firefox, Internet explorer with the latest version the user can access the facility of the web portal.



7. Limitation And Future Scope

Limitations:

Some of the limitations and key points of further enhancement can be listed out as follows –

- This software is used to provide the information about all Products truncation.
- Customer can read information about all products.
- Admin can enter details related to a products.

8. Security Mechanism

Security means different things to different people depending upon their perspective. In the context of our application it means security of the data from unauthorized access and modification i.e. only authorized user should be able to view presented information according to their access permissions.

9. Bibliography

Reference Book:

- Core Java ™ 2 Volume II – By Cay Horstmann, Gary Cornell
- J2EE by GARRY CORNELL.
- J2EE by Allamaraju & Buest (Apress Publications).
- JSP & Servlets by Marty Hall (SUN Microsystems)
- SYSTEM ANALYSIS AND DESIGN by ELIAS M AWAD.
- SOFTWARE ENGINEERING by IAN SOMMEVILLE.
- SOFTWARE ENGINEERING by Pankaj Jalote.
- SOFTWARE ENGINEERING by PRESSMAN.
- THINKING IN JAVA by BRUCE ECKEL.

Website:

- http://www.sun.com
- http://www.coreservlets.com
- http://www.serverside.com
- http://www.w3schools.com
- http://www.google.com
- http://www.wikipedia.com
- http://www.webopedia.com
- http://www.ddj.com
- http://www.java-source.net
- http://www.tripadvisor.com