**Gujarat Technological University, Ahmedabad**

**SARDAR VALLABHBHAI PATEL INSTITUTE OF TECHNOLOGY, VASAD**

**2019-2020**

**A PROJECT REPORT ON**

**IT-HELPDESK AND ASSET MONITORING**

*Submitted by*

**Krishna Patel [160410107082]**

**Khanjan Shah [160410107117]**

***In partial fulfilment Of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER ENGINEERING GUIDED BY:**

**Assit. Prof. Jayna Shah**

**SARDAR VALLABHBHAI PATEL INSTITUTE OF TECHNOLOGY, VASAD**

**COMPUTER ENGINEEERING DEPARTMENT**



**CERTIFICATE**

**Date: 25-04-2020**

**This is to certify that the project entitled “IT-HELPDESK AND ASSET MONITORING” has been carried out by Krishna Patel(160410107082), Khanjan Shah(160410107117), under my guidance in partial fulfilment of the project in Bachelor Of Engineering in Computer Engineering 8th semester of Gujarat Technological University, during the academic year 2019-2020.**

**Internal Guide: Head of Department:**

**Prof. Jayna Shah Prof. Dr. Neha Soni**

**Assistant Professor C.E. Department**

**S.V.I.T. Vasad S.V.I.T. Vasad**

**ACKNOWLEDGEMENT**

We would like to thank our project internal guide Prof. Jayna Shah, (Computer engineering department SVIT, Vasad) for providing a vision about the project. We have been greatly benefited from the well-organized critical reviews and inspiration throughout our work.

We would also like to thank our faculty members for their unfailing cooperation and their valuable time to assist us in our work. We have developed not only technical skills but also learned all those qualities required to become a good professional engineer.

Last but not least, we would like to mention here that we are greatly thankful to each and everybody who has been associated with this project at any stage but whose name does not find a place in this acknowledgement.

**ABSTRACT**

One of the key investments in every organization are the assets. Systems for handling data about assets are valuable investment as well. Being able to have an overview of the assets, information about every detail regarding them and moreover being able to analyze that data can be critical for the business. The subject of this project report is to show the importance of the asset management strategy, point the key role that an asset management system has and list out the desired features. Another important aspect of the project report is to show that Software as a Service (SaaS) business model is best suitable model, especially for small and medium companies. This project report tries to define a methodology for evaluating asset management systems. Some of the existing solutions are being evaluated against it and compared with each other.

As we know, sometimes due to natural disasters employees are unable to visit to their office for work and on telephone also it is not possible to manage the assets , due to this irregularities there are many loses faced by company every year.

Due to the problems faced by employees and to solve this drawbacks, we made a web- application as per our project definition. In this web-application, assets are monitored and managed online by generating token-id of each assets. Employees have allocated their own login and asset id, and if any problem arise they have to contact directly to the engineer for repairing assets. Every employee can place ticket in helpdesk for their problem.

Then the complain goes to service engineer as SMS and GPS locations of employee (Vadodara, Savli, Godhra, Dabhoi, Dahod etc…). (AMC) payment would be done on the basis of the reports.\* AMC contractor must attain the call in 24 hours, otherwise they will be charged penalty for this.

**Contents**

|  |  |  |
| --- | --- | --- |
| **1.** | **Introduction……………………………………………………………………………** | **8** |
| 1.1 | [Project Summary](#_bookmark0)………………………………………………………………………. | 8 |
| 1.1.1 | Why you made this project…………………………………………………………. | 8 |
| 1.2 | Project Details…………………………………………………………………………. | 8 |
| 1.2.1 | Where your project or innovation will use…………………………………………. | 8 |
| 1.2.2 | How it works……………………………………………………………………….. | 8 |
| 1.2.3 | What it requires…………………………………………………………………….. | 8 |
| 1.3 | Project Specification…………………………………………………………………… | 9 |
| 1.3.1 | The main work of project…………………………………………………………... | 9 |
| 1.4 | Literature Review and Prior Art Search……………………………………………….. | 9 |
| 1.4.1 | From where you search and learn this……………………………………………… | 9 |
| 1.4.2 | Patent overview(PSAR)……………………………………………………………. | 9 |
| 1.5 | Plan of project…………………………………………………………………………. | 11 |
| **2.** | **Requirement Specification……………………………………………………………** | **14** |
| 2.1 | Functional Requirement……………………………………………………………….. | 14 |
| 2.2 | [Non-Functional Requirement](#_bookmark13)………………………………………………………….. | 15 |
| 2.3 | Hardware Requirement………………………………………………………………… | 16 |
| 2.4 | Software requirement…………………………………………………………………. | 16 |
| 2.5 | Feasibility Study………………………………………………………………………. | 17 |
| 2.5.1 | Economic Feasibility………………………………………………………………. | 17 |
| 2.5.2 | Technical Feasibility……………………………………………………………….. | 18 |
| **3** | **System Design………………………………………………………………………....** | **19** |
| 3.1 | System Diagrams…………………………………………………………………….... | 19 |
| 3.1.1 | Activity Diagram…………………………………………………………………… | 19 |
| 3.1.2 | Sequence Diagram……………………………………………………………......... | 20 |
| 3.1.3 | Class Diagram………………………………………………………………………. | 21 |
| 3.1.4 | Use Case Diagram………………………………………………………………….. | 22 |
| 3.1.5 | E-R Diagram……………………………………………………………………….. | 23 |
| 3.1.6 | Data Dictionary…………………………………………………………………….. | 24 |
| **4.** | **System Description……………………………………………………………………** | **27** |
| 4.1 | Software Description (Visual Studio)…………………………………………………. | 27 |
| **5.** | **Implementation……………………………………………………………………….** | **28** |
| 5.1 | Coding Standards……………………………………………………………………… | 28 |
| 5.2 | User Interface………………………………………………………………………..... | 29 |
| **6.** | **Conclusion & Future Scope………………………………………………………….** | **37** |
| **7.** | **Canvases and Modules……………………………………………………………….** | **38** |
| 7.1 | AEIOU Summary……………………………………………………………………… | 38 |
| 7.2 | Empathy Mapping Canvas…………………………………………………………….. | 39 |
| 7.3 | Ideation Canvas……………………………………………………………………....... | 40 |
| 7.4 | Product development Canvas…………………………………………………………. | 41 |
| 7.5 | Business Model Canvas……………………………………………………………….. | 42 |
| **8.** | **References……………………………………………………………………………..** | **43** |
| **9.** | **Plagiarism Report…………………………………………………………………….** | **44** |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Fig.3.1.1 | Activity Diagram……………………………………………………………….. | 19 |
| Fig.3.1.2 | Sequence Diagram……………………………………………………………… | 20 |
| Fig.3.1.3 | Class Diagram………………………………………………………………….. | 21 |
| Fig.3.1.4 | Use Case Diagram……………………………………………………………… | 22 |
| Fig.3.1.5 | ER-Diagram……………………………………………………………………. | 23 |
| Fig.5.2.1 | Login Page…..…………………………………………………………………. | 29 |
| Fig.5.2.2 | Index Page……………………………………………………………………… | 30 |
| Fig.5.2.3 | Employee Registration Page…………………………………………………… | 30 |
| Fig.5.2.4 | Item Master Page………………………………………………………………. | 31 |
| Fig.5.2.5 | Assets Page…………………………………………………………………….. | 31 |
| Fig.5.2.6 | Asset Allocation Page………………………………………………………….. | 32 |
| Fig.5.2.7 | AMC Engineer Registration Page……………………………………………… | 33 |
| Fig.5.2.8 | Problem Category Page………………………………………………………… | 33 |
| Fig.5.2.9 | Generate Ticket Page…………………………………………………………... | 34 |
| Fig.5.2.10 | Ticket Details Page…………………………………………………………….. | 34 |
| Fig.5.2.11 | Engineer Login page…………………………………………………………… | 35 |
| Fig.5.2.12 | Location Engineer Ticket Details Page…………………………………………. | 36 |
| Fig.7.1 | AEIOU Summary………………………………………………………………. | 38 |
| Fig.7.2 | Empathy Mapping Canvas……………………………………………………… | 39 |
| Fig.7.3 | Ideation Canvas………………………………………………………………… | 40 |
| Fig.7.4 | Product Development…………………………………………………………... | 41 |
| Fig.7.5 | Business Model Canvas………………………………………………………… | 42 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| Table: 2.1 | Functional Requirements…………………………………………………... | 14 |
| Table: 2.2 | Non-Functional requirement………………………………………………. | 15 |
| Table: 3.1.6.1 | Employee Database………………………………………………………... | 24 |
| Table: 3.1.6.2 | Asset Database……………………………………………………………... | 24 |
| Table: 3.1.6.3 | Asset Allocation Database…………………………………………………. | 24 |
| Table: 3.1.6.4 | Ticket Database……………………………………………………………. | 25 |
| Table: 3.1.6.5 | AMC Vendor Database……………………………………………………. | 25 |
| Table: 3.1.6.6 | AMC Order Database……………………………………………………… | 25 |
| Table: 3.1.6.7 | AMC Engineer Database…………………………………………………... | 26 |

**Chapter 1: Introduction**

# Project Summary

* + 1. **Why you made this project?**

As we know, sometimes due to natural disasters employees are unable to visit to their office for work and on telephone also it is not possible to manage the assets , due to this irregularities there are many loses faced by company every year.

So, we planned to make this project for the employees and help them to reach to their planned target given by the company.

# Project Details

* + 1. **Where your project or innovation will use?**

As it is a live project of MGVCL(GEB) and it is a business project so is used in MGVCL’s sectors only, here the employers are the only customers of MGVCL.

* + 1. **How it works?**

In this web-application, assets are monitored and managed online by generating token-id of each assets. Employees have allocated their own login and asset id, and if any problem arise they have to contact directly to the engineer for repairing assets.

Every employee can place ticket in helpdesk for their problem. Then the complain goes to service engineer as SMS and GPS locations of employee( Vadodara, Savli, Godhra, Dabhoi, Dahod etc…)

(AMC) payment would be done on the basis of the reports. AMC contractor must attain the call in 24 hours, otherwise they will be charged penalty for this.

* + 1. **What it requires?**

High speed internet to access the token id.

Assets required by MGVCL sectors.

# Project Specification

* + 1. **The main work of project**

We planned to make this project for the employees and help them to reach to their planned target given by the company.

Now this is a project for MGVCL, we are making a web-application to monitor the assets of industry and now it is easy to online featuring the assets so that employees can do their job by online if they are not able to come to the office for genuine reason.

# Literature review and Prior art Search

* + 1. **From where you search and learn this?**

We contact online to the mgvcl manager to learn about this project and we had to know about it through mgvcl website, as they upload prologues for this project. We read it and search it on internet and find it useful for business activity and so we planned to do this project. ([www.mgvcl.com](http://www.mgvcl.com/)), additional google patents.

* + 1. **Patent Overview (PSAR).**

To learn more about this project that how we start implementing and what is required to implement it we searched for some related patents through google and find required activity for this project.

The patents we use for this project are:

**To Display status of an asset using stand-alone software system.**

A system and method of monitoring assets of an enterprise using a stand-alone software system and a process automation software system are provided. The stand-alone system is operable to generate a web page about a condition of the asset and to transmit data items for the condition in a single data string.

In accordance with the present invention, a method is provided for use in an enterprise having an asset. In accordance with the method, data is gathered about at least one condition of the asset. A status of the at least one condition is determined from the gathered data. For each

condition, a web page is generated containing information about the condition and data items are transmitted in a single data string.

While the invention has been shown and described with respect to particular embodiments thereof, those embodiments are for the purpose of illustration rather than limitation. Other variations and modifications of the specific embodiments herein described will be apparent to those skilled in the art, all within the intended spirit and scope of the invention.

To improve the condition of this stand-alone system we have to take following point under consideration:

Yes, instead of generating web-page for each condition of an asset we have to make one single web-page that contains whole information about assets but in such a way that on a single web- page there are different keys to operate that asset.

**A virtual token (general requirements of mgvcl)**

A virtual token represents an item, and includes embedded data defining rules and/or capabilities which apply to the use of the item. A virtual token may include graphical image data which is used to generate a display on a computer, whereby selection of the display allows the item represented by the virtual token to be used.

The present invention relates to a virtual token and to methods and systems for providing and using virtual tokens. Electronic commerce is touted as being the basis on which the majority of all transactions will be conducted in the future. For it to be accepted and utilised requires a change in traditional concepts associated with value, how it relates to assets or services.

As basic assets and services become commoditised (especially digital assets or services) and economic value shifts towards personalised packages of assets and services, the value of a given offering will differ by content, individual, time, and circumstance. \* Electronic money, barter tokens, e-wallets, online auctions, and online trading initiatives are currency-centric and do not address this issue.

Key Learning points.

Virtual token including identifying data, data defining a presentation of said virtual token, and instructions for sending an access request to another computer device, etc.

# Plan of project

To start implementing the project, first we planned for this project that how we start and from where we start this. So first of all we collect the information about this project so that we can make it easy to implement further.

Next we design some working diagrams by fully understanding the project and make flowcharts, activity diagrams etc. Next we start collecting and making databases that are required for this project and further implementation is still going on.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | June-July | August | September | October |
| Title |  |  |  |  |
| Objective analysis |  |  |  |  |
| Diagrams & Data-Dict |  |  |  |  |
| Canvases & Modules |  |  |  |  |
| Primary report |  |  |  |  |
| Start implementing |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Nov-Dec | January | February | March |
| Login Credentials |  |  |  |  |
| Database Design |  |  |  |  |
| GUI Design |  |  |  |  |
| Module Analysis |  |  |  |  |
| Module implementation |  |  |  |  |
| Final Layout |  |  |  |  |

**Initial idea**



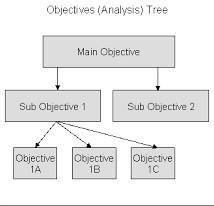
Here we describe the idea about how we get to know about this project and what idea we would apply to initialize this project.

So we get this project idea from the MGVCL website and we first decided to get circumstances based on this project. We started collecting the information that at what places this project use and in how it is benefited to us and industry.

We visit to MGVCL company and discuss there about this project its use, advantages and disadvantages. Collecting all information we next proceed to our internal guides to discuss and explain this project and they also suggest us some better idea for this project and next we started searching patents for this project so that we come to know how we should implement it and how much is necessary to implement this project because there are already some systems exists for project so we try to implement them through our new innovative ideas.

**Ob………**

Here we give the analysis of objects that we used in this project and how we used it.



So for this project we have to build a Web-Application using a DOTNET(MVC) technology that is our main objective. The implementation takes around 4 months from starting date and before that we have to finished our analysis and paper work, we have to prepare our diagrams like: Activity diagram, use case, Sequence flow diagram, flow chart, and data dictionary.

**Chapter 2: Requirement Specification**

# Functional Requirement

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used.

Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail.

**Table: 2.1: Functional Requirements**

|  |  |
| --- | --- |
| Functional Requirements Number | Functional Requirement Description |
| FR-1 | Registration should be done by Admin |
| FR-2 | Admin should have to create Users |
| FR-3 | Asset id should be generated for engineers by input users |
| FR-4 | Assets should be allocated to end-users |
| FR-5 | Records of repaired assets should be maintained by admin |
| FR-6 | Tickets should be generated to repair assets by end users |
| FR-7 | Ticket status should be regularly updated by engineers |
| FR-8 | Registration for AMC should be done to admin |
| FR-9 | Quater bill should be generated after finishing contract by AMC |

# Non-Functional Requirement

Basically, Non-functional requirements describe how the system works**,** while functional requirements describe what the system should do**.**

This does not mean the latter are more important, but most requirement gathering techniques focus on functional requirements**,** so large gaps in non-functional requirements are common.

So what exactly are we looking for here? Well, here are four examples of Non- Functional requirement groups**;** usability, reliability, performance, and supportability, as well as a few top tips on each one.

**Table: 2.2 Non-Functional requirement**

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| Registration | Mandatory field that collects required data of registered person. |
| Login | Secured step where registered person can accessed with id & password. |
| Token-id | Gives access to assets which are required to repair. |
| Assets | Include hardware components like Keyboard, router, mouse, digital meter |
| Record book | Keeps record of regularly updated information |
| Quarter bill | Gives estimation of cost for full contract |
| Admin | Handle whole application with regular updates |
| AMC | Annual maintenance contractor generates quarter bill |

# Hardware Requirement

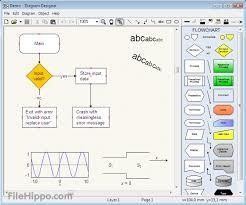
**Hardware:** 500MB Hard Disk space for installation. Recommended minimum CPU- 3.2GHZ

Recommended 1GB RAM for a central server with 3 nodes.

# Software Requirement

**Software:** Visual studio latest version with MVC tools and requirements Diagram designer to draw diagrams.

Excel sheet to keep record of cost & estimation. MS-PowerPoint to present the project demo.



# Feasibility Study

* + 1. **Economic feasibility Cost**

The cost for this project will not go too much high, because it is totally a software base project, but the maintenance charges for this project may applied.

* + - * The charge for regular update of information.
      * The charge for generating token-id for assets.
      * The charge for using industry’s software and hardware components.

**Value**

Applying such cost to implement the project it gives benefits like:

* + - * Ease of doing work as it is online and daily updated.
      * Knowing about latest tokens for assets through token-id.
      * Smooth and easy use of software experience from industry.
      * Best & affordable hardware components as required for project.



# Technical feasibility

**What is technical feasibility?**

Technical feasibility study assesses the details of how you will deliver a product or service (i.e., materials, labor, transportation, where your business will be located, technology needed, etc.).

Think of the technical feasibility study as a logistical or tactical plan of how your business will produce, store, deliver and track its products or services.

**Technical feasibility**

The only risk for using this project idea is that without internet connectivity the work is incomplete and also chances for information lost if in between the work internet connectivity lost. So when assets are delivered with their token-id, at that time there is possibility of losing token-id due to poor network connection.

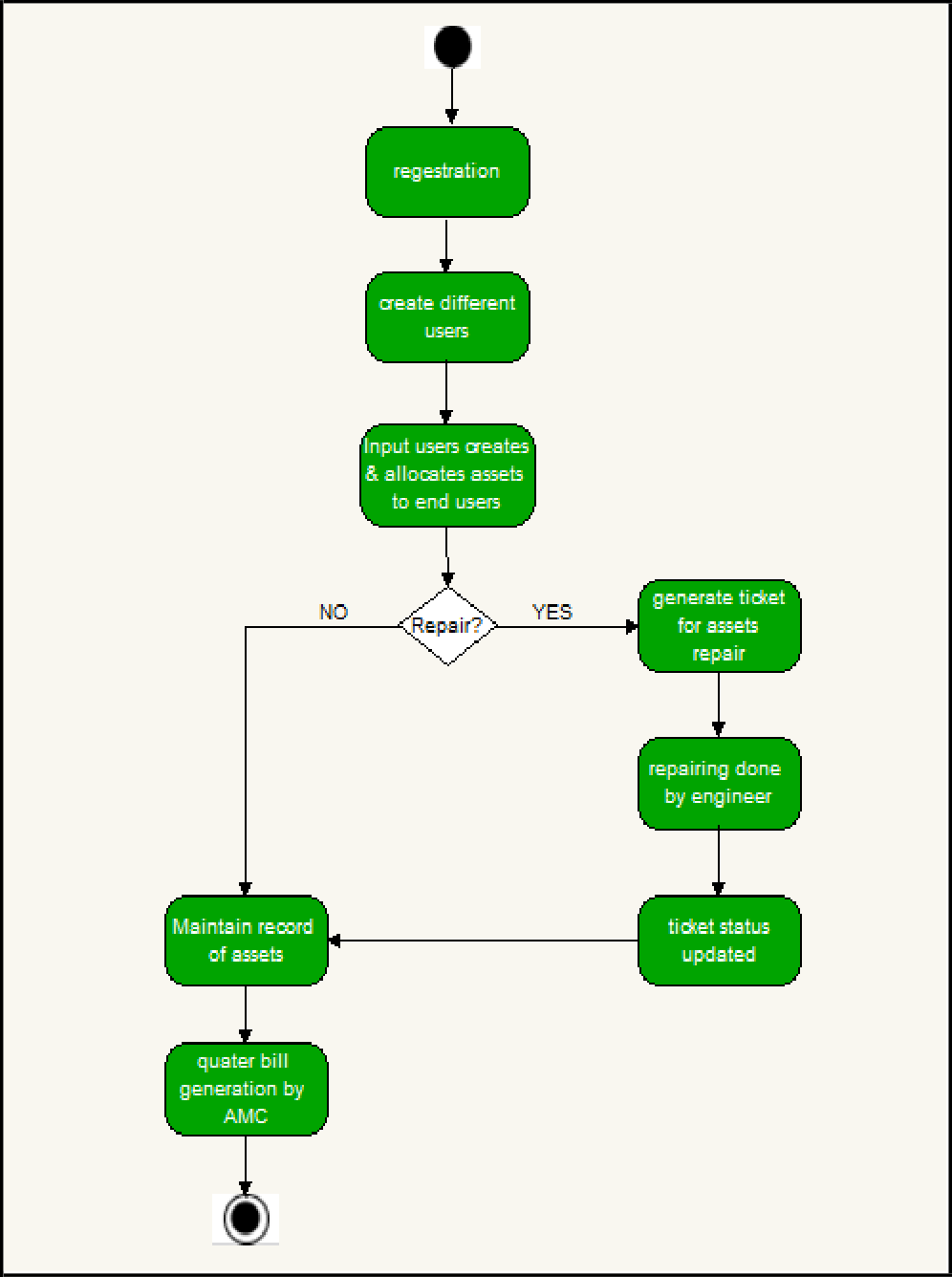
**Solution**

We have provided facility for WAN-connection, so that if sometime network is poor we can use it through VPN also. And assets are delivered as per their accessing time so if network loss, there is no matter of concern to think about where it is because without token-id it can not move further and came back to the source station.

**Chapter 3: System Design:**

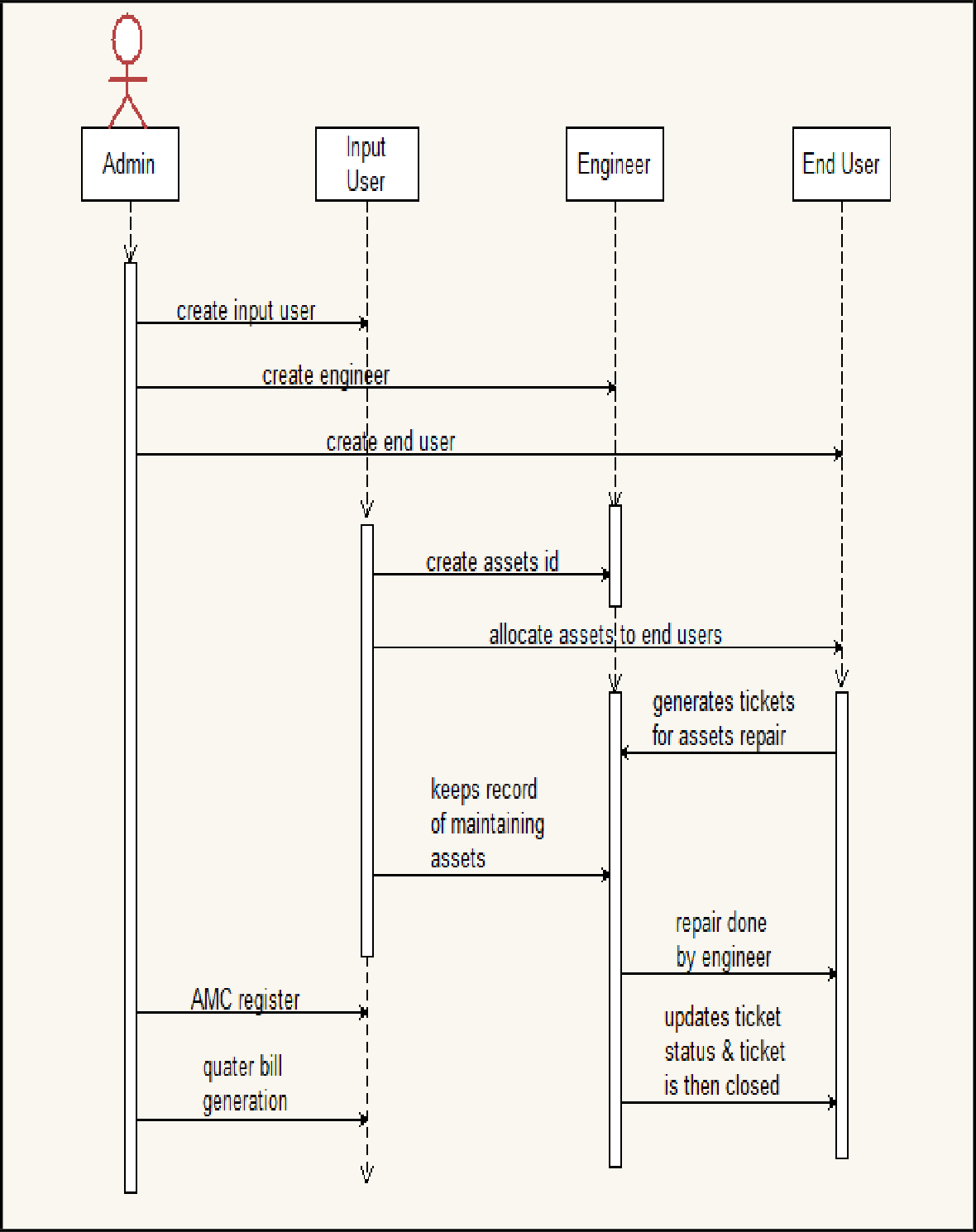
# System Diagrams

# Activity diagram



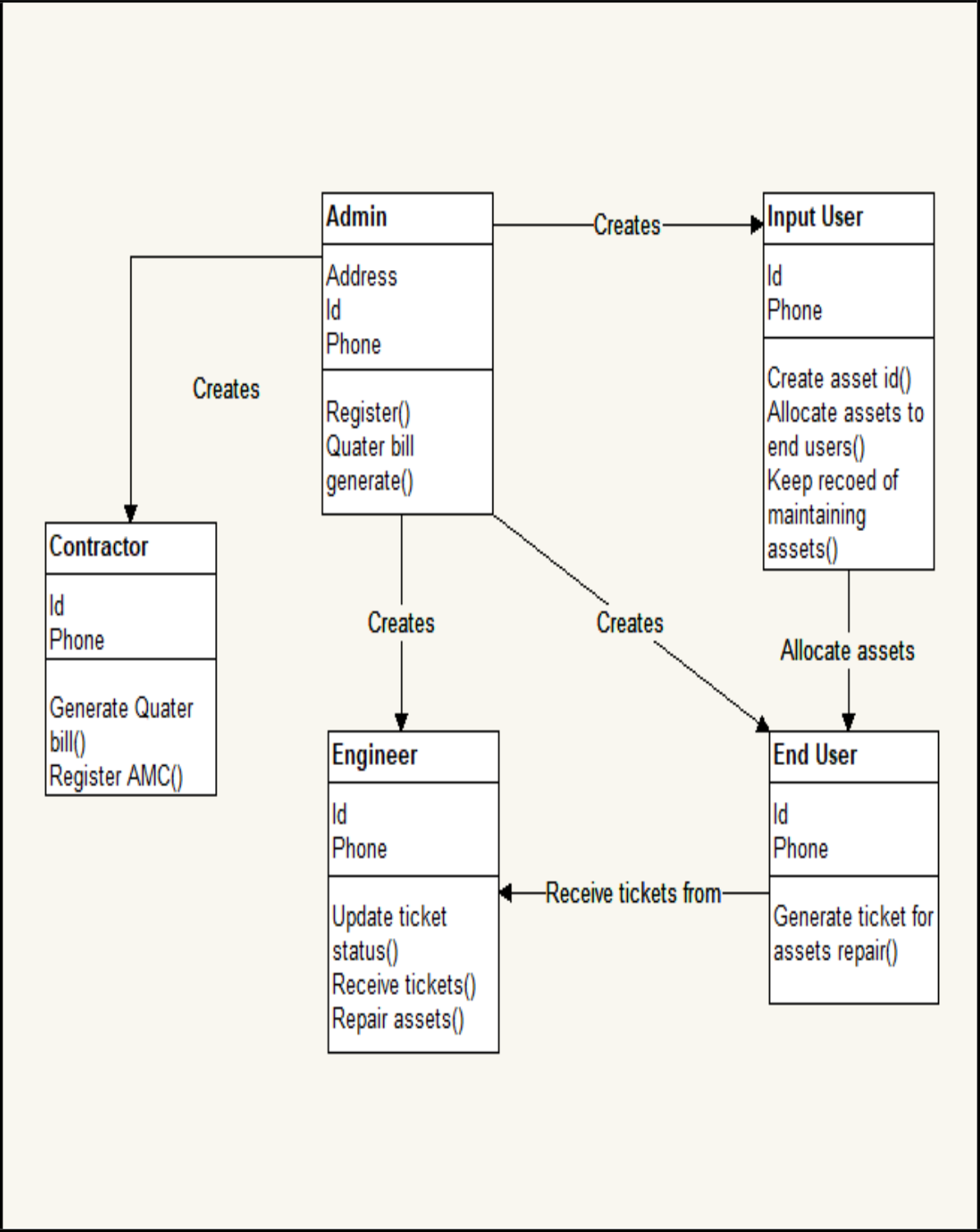
**Fig.3.1.1 Activity Diagram**

# Sequence diagram



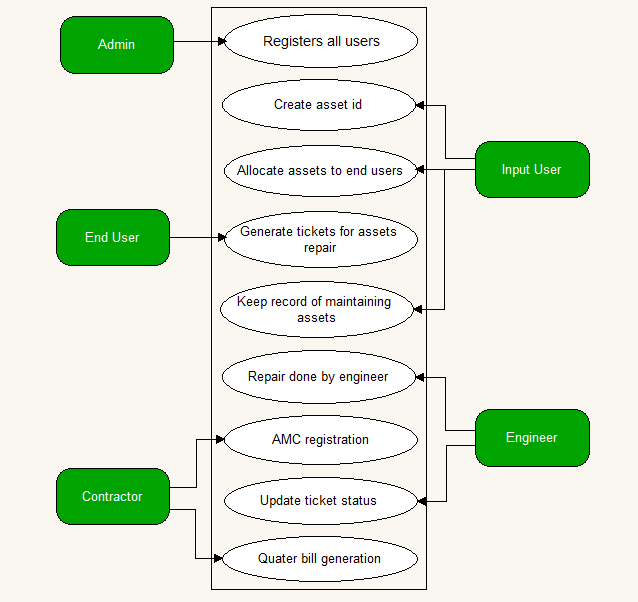
**Fig.3.1.2 Sequence Diagram**

# Class diagram



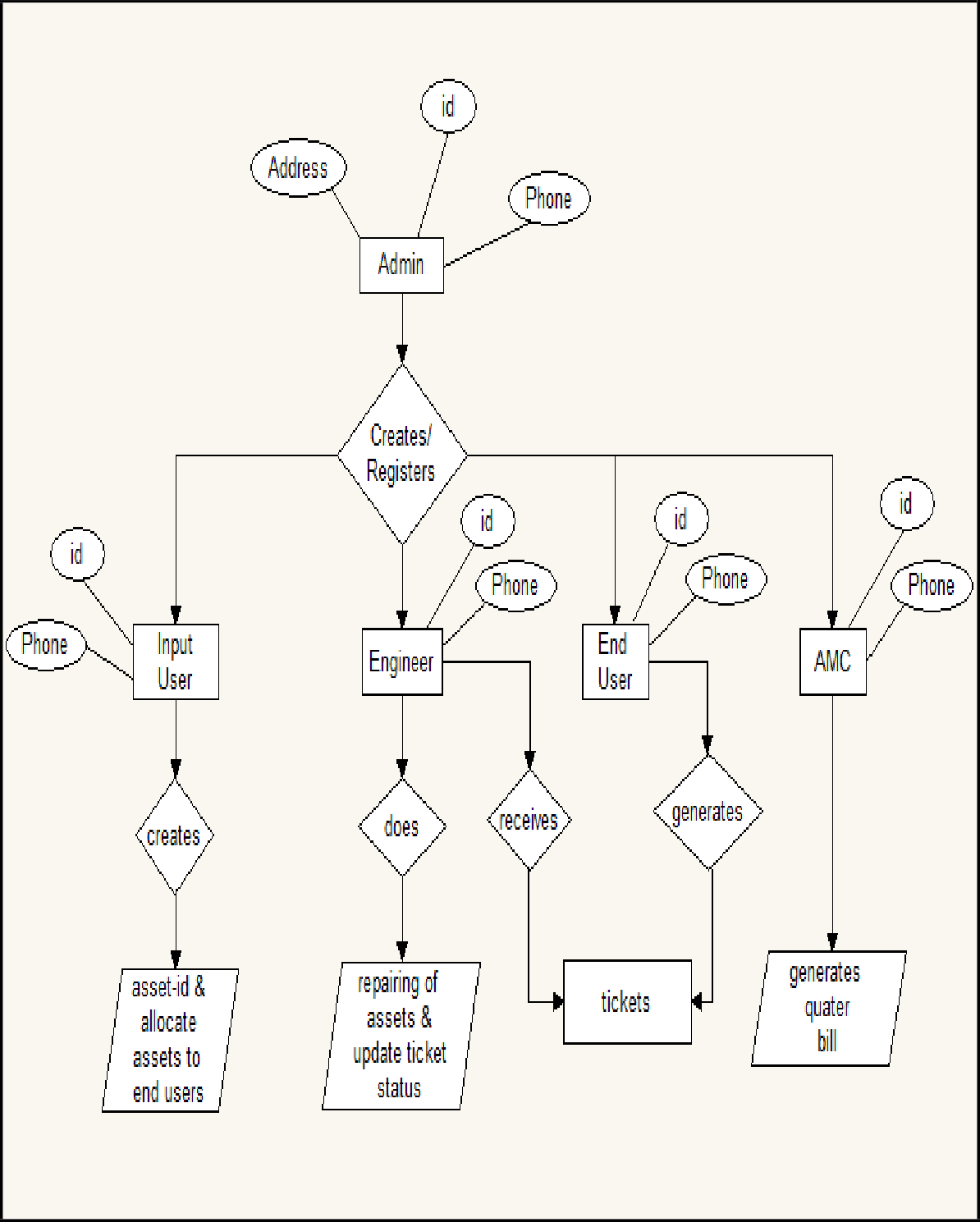
**Fig.3.1.3 Class Diagram**

# Use Case



**Fig.3.1.4 Use Case**

# ER- diagram



**Fig.3.1.5 ER-Diagram**

# Data Dictionary

**Table 3.1.6.1: Employee Database**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** |
| EmpNo | Varchar (50) | Primary Key |
| Password | Varchar(20) | Not Null |
| Name | Varchar(20) | Not Null |
| Gender | Bit | Not Null |
| Email | Varchar(20) | Not Null |
| Designation | Char(20) | Not Null |
| Mobile | Numeric(20,0) | Not Null |
| Location | Varchar(20) | Not Null |
| Department | Varchar(20) | Not Null |
| Floor | Varchar(20) | Not Null |
| Type | Char(20) | Not Null |

**Table 3.1.6.2: Asset Database**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** |
| Asset\_Id | Varchar(20) | Primary key |
| Id | Numeric(18,0) | Not Null |
| Company | Varchar(50) | Not Null |
| Model | Varchar(50) | Not Null |
| SerialNumber | Varchar(50) | Not Null |
| PurchaseDate | Date | Not Null |
| Location | Varchar(50) | Not Null |
| Status | Char(50) | Not Null |

**Table 3.1.6.3: Asset Allocation Database**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** |
| Id | int | Primary key |
| EmpNo | Varchar(50) | Not Null |
| Computer | Varchar(20) | Not Null |
| Mouse | Varchar(20) | Not Null |
| Laptop | Varchar(20) | Not Null |
| Printer | Varchar(20) | Not Null |
| Router | Varchar(20) | Not Null |

**Table 3.1.6.4: Ticket Database**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** |
| Ticket\_No | Int | Primary Key |
| Empno | Varchar(10) | Not Null |
| Asset | Varchar(50) | Not Null |
| Asset\_Id | Varchar(50) | Not Null |
| Subject | Varchar(50) | Not Null |
| Problem Category | Varchar(50) | Not Null |
| DetailDescription | Varchar(50) | Not Null |
| File | Varbinary(MAX) | Not Null |
| OpenDate | Date | Not Null |
| CloseDate | Date | Not Null |
| Status | Char(50) | Not Null |
| Location | Varchar(50) | Not Null |

**Table 3.1.6.5: AMC Vendor Database**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** |
| VendorNumber | Varchar(50) | Primary Key |
| VendorName | Varchar(50) | Not Null |
| Address | Varchar(50) | Not Null |
| Mobile | Numeric(20,0) | Not Null |
| Email | Varchar(50) | Not Null |
| GSTNumber | Varchar(50) | Not Null |
| Location | Varchar(50) | Not Null |

**Table 3.1.6.6: AMC Order Database**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** |
| AMCOrderNumber | Varchar(50) | Primary Key |
| VendorName | Varchar(50) | Not Null |
| PurchaseOrderNumber | Varchar(50) | Not Null |
| OutwardNumber | Varchar(50) | Not Null |
| AMCStartDate | Date | Not Null |
| AMCEndDate | Date | Not Null |
| ContractValue | Varchar(50) | Not Null |
| Location | Varchar(50) | Not Null |

**Table 3.1.6.7: AMC Engineer Database**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Constraint** |
| Id | Int | Primary Key |
| AMCOrderNumber | Varchar(50) | Not Null |
| Number of Engineers | Varchar(50) | Not Null |
| EngineerName | Varchar(50) | Not Null |
| Email | Varchar(50) | Not Null |
| Mobile | Numeric(20,0) | Not Null |
| Address | Varchar(50) | Not Null |
| Location | Varchar(50) | Not Null |
| Password | Varchar(20) | Not Null |

**Chapter 4: System Description.**

# 4.1 Software description (Visual studio).

Microsoft Visual Studio is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs,](https://en.wikipedia.org/wiki/Computer_program) as well as [websites,](https://en.wikipedia.org/wiki/Web_site) [web apps,](https://en.wikipedia.org/wiki/Web_app) [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps.](https://en.wikipedia.org/wiki/Mobile_app) Visual Studio uses Microsoft software development platforms such as [Windows API](https://en.wikipedia.org/wiki/Windows_API), [Windows](https://en.wikipedia.org/wiki/Windows_Forms) [Forms,](https://en.wikipedia.org/wiki/Windows_Forms) [Windows Presentation Foundation,](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation) [Windows Store](https://en.wikipedia.org/wiki/Windows_Store) and [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight). It can produce both [native code](https://en.wikipedia.org/wiki/Machine_code) and [managed code](https://en.wikipedia.org/wiki/Managed_code).

Visual Studio includes a [code editor](https://en.wikipedia.org/wiki/Code_editor) supporting [IntelliSense](https://en.wikipedia.org/wiki/IntelliSense) (the [code completion](https://en.wikipedia.org/wiki/Code_completion) component) as well as [code refactoring.](https://en.wikipedia.org/wiki/Code_refactoring) [The integrated debugger](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) works both as a source-level debugger and a machine-level debugger. Other built-in tools include a [code profiler,](https://en.wikipedia.org/wiki/Profiling_(computer_programming)) designer for building [GUI](https://en.wikipedia.org/wiki/GUI) applications, [web designer,](https://en.wikipedia.org/wiki/Web_designer) [class](https://en.wikipedia.org/wiki/Class_(computing)) designer, and [database schema](https://en.wikipedia.org/wiki/Database_schema) designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for [source](https://en.wikipedia.org/wiki/Source_control) [control](https://en.wikipedia.org/wiki/Source_control) systems (like [Subversion](https://en.wikipedia.org/wiki/Subversion_(software)) and [Git](https://en.wikipedia.org/wiki/Git)) and adding new toolsets like editors and visual designers for [domain-specific languages](https://en.wikipedia.org/wiki/Domain-specific_language) or toolsets for other aspects of the [software development](https://en.wikipedia.org/wiki/Software_development_lifecycle) [lifecycle](https://en.wikipedia.org/wiki/Software_development_lifecycle) (like the [Azure DevOps](https://en.wikipedia.org/wiki/Azure_DevOps_Server) client: Team Explorer).

Visual Studio supports 36 different [programming languages](https://en.wikipedia.org/wiki/Programming_language) and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language- specific service exists. Built-in languages, [Visual Basic](https://en.wikipedia.org/wiki/Visual_Basic_.NET)

[.NET](https://en.wikipedia.org/wiki/Visual_Basic_.NET), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [F#](https://en.wikipedia.org/wiki/F_Sharp_(programming_language)), [JavaScript,](https://en.wikipedia.org/wiki/JavaScript) [TypeScript,](https://en.wikipedia.org/wiki/TypeScript) [XML,](https://en.wikipedia.org/wiki/XML) [XSLT](https://en.wikipedia.org/wiki/XSLT), [HTML](https://en.wikipedia.org/wiki/HTML), and [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). Support for other languages such as [Python,](https://en.wikipedia.org/wiki/Python_(programming_language)) [Ruby,](https://en.wikipedia.org/wiki/Ruby_(programming_language)) [Node.js,](https://en.wikipedia.org/wiki/Node.js) and [M](https://en.wikipedia.org/wiki/MUMPS) among others is available via [plug-ins.](https://en.wikipedia.org/wiki/Plug-in_(computing)) [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) (and [J#](https://en.wikipedia.org/wiki/J_Sharp)) were supported in the past.

The most basic edition of Visual Studio, the Community edition, is available free of charge. The slogan for Visual Studio Community edition is "Free, fully-featured IDE for students, open-source and individual developers". The currently supported Visual Studio version is 2019.

Initially referred to as Visual Studio "14", the first [Community Technology Preview](https://en.wikipedia.org/wiki/Community_Technology_Preview) (CTP) was released on 3 June 2014 and the [Release Candidate](https://en.wikipedia.org/wiki/Release_Candidate) was released on 29 April 2015; Visual Studio 2015 was officially announced as the final name on 12 November 2014.

Visual Studio 2015 RTM was released on 20 July 2015. Visual Studio 2015 Update 1 was released on 30 November 2015. Visual Studio 2015 Update 2 was released on 30 March 2016. Visual

Studio 2015 Update 3 was released on 27 June 2016.

**Chapter 5: Implementation**

# Coding Standards:

* + - A coding standard gives a uniform appearance to the codes written by different engineers.
    - It improves readability and maintainability of the codes and it reduces complexity also.
    - It helps in codes reuse and helps to detect error easily.
    - It promotes sound programming practices and increases efficiency of the programmers.

**Standard headers for different modules:**

For better understanding and maintenance of the code, the header of different module should follow some standard format information

* + - * Naming conventions for local variables, global variables, constants and functions. The names of the function should be written in camel case starting with small letters.
      * The name of the functions must describe the reason of using the functions clearly and briefly.

**Indentation:**

Proper indentation is very important to increase the readability of the code. For making the code readable, programmers should use white space properly. Some of the spacing conventions are given below:

**Code should be well documented:**

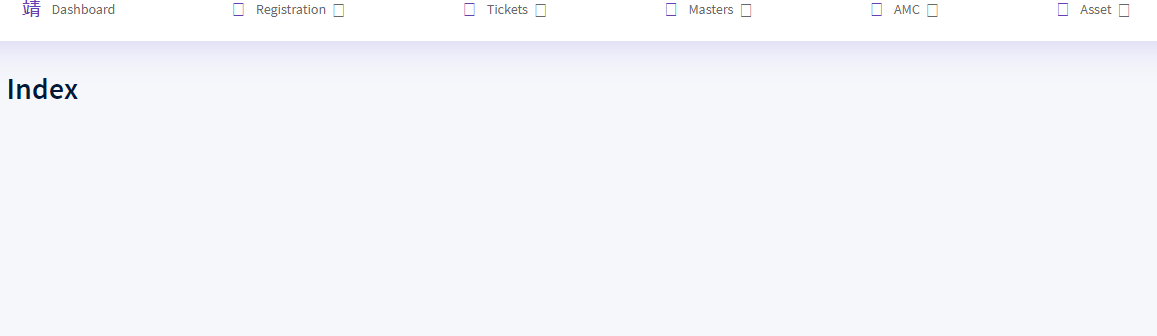
**Length function should not be very large:**

Lengthy function are very difficult to understand. That’s why function should be small work and lengthy functions should be broken into small ones for completing small tasks.

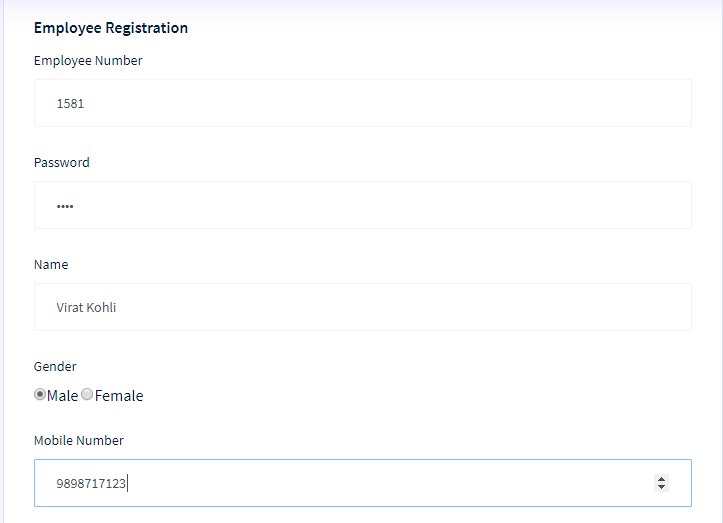
# User Interface



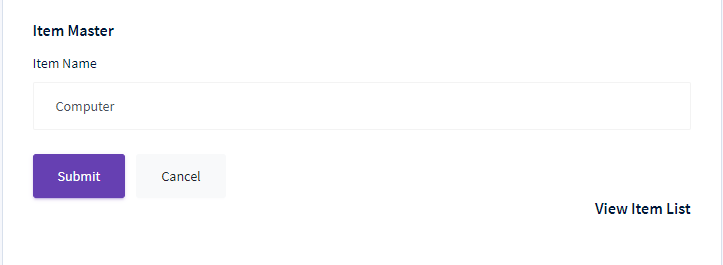
**Fig.5.2.1 Login Page**



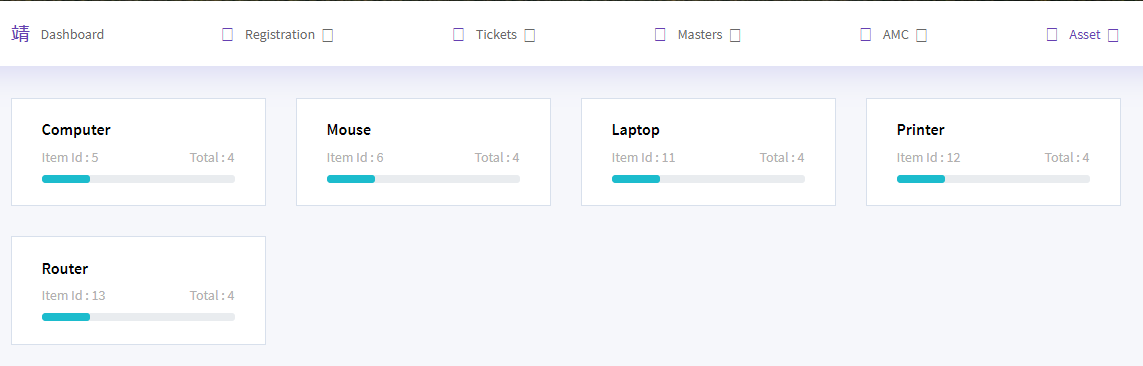
**Fig.5.2.2 Index Page**



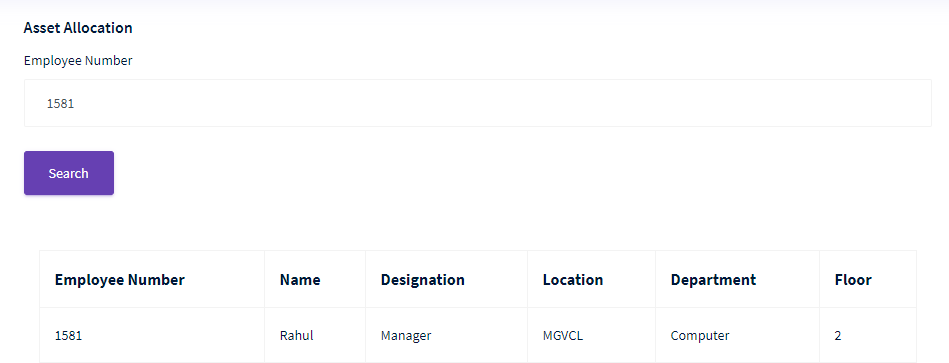
**Fig.5.2.3 Employee Registration Page**



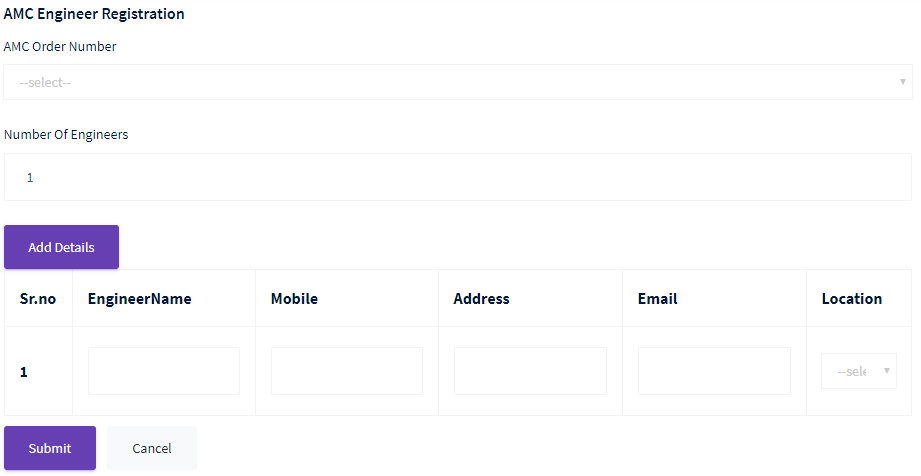
**Fig.5.2.4 Item Master Page**



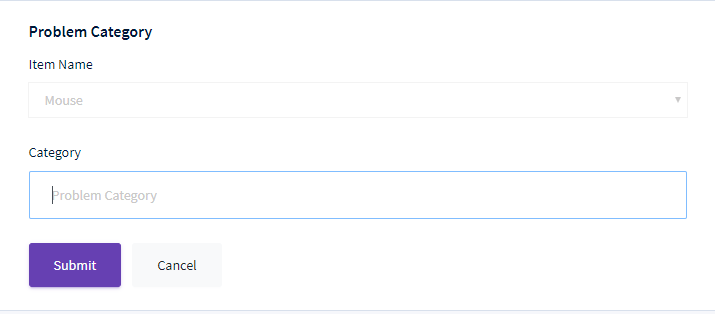
**Fig.5.2.5 Assets Page**



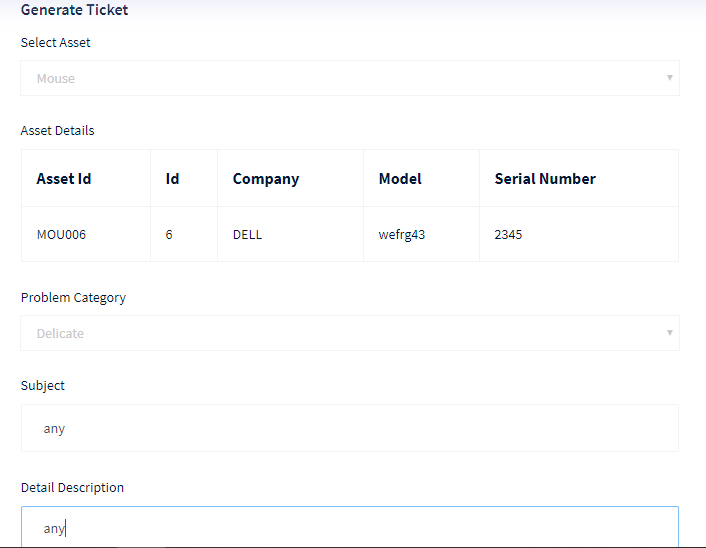
**Fig.5.2.6 Asset Allocation Page**



**Fig.5.2.7 AMC Engineer Registration Page**



**Fig.5.2.8 Problem Category Page**



**Fig.5.2.9 Generate Ticket Page**

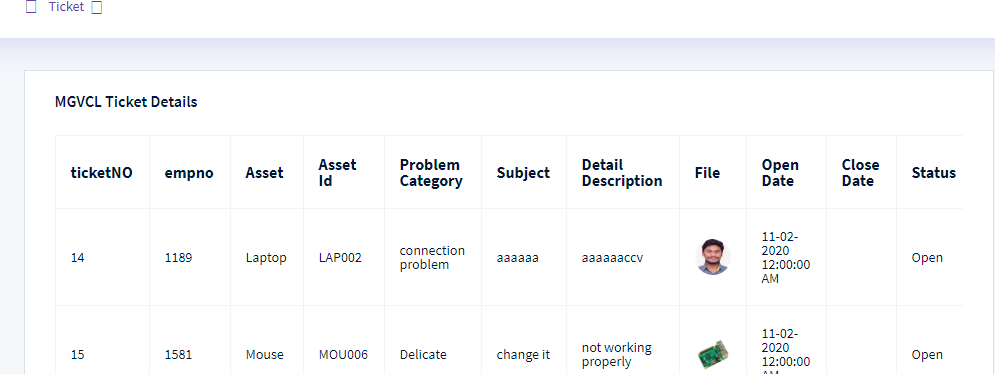


**Fig.5.2.10 Ticket Details Page**

**Engineer Login:**



**Fig.5.2.11 Engineer Login Page**



**Fig.5.2.12 Location Engineer Ticket details Page**

**Chapter 6: Conclusion & Future scope**

Benefits from asset management strategy appliance are obvious. Also having a good asset management system is one of the most important aspects. In this project report we tried to make an overview of desired features and technical preconditions. Also we tried to value our desirability of such functions and value some of the existing software packages against our methodology. General conclusion is that some of the systems managed to gain high scores in some of the sections excelling in some of the features but failed in the rest of the metrics. Having in mind the need of such systems gives us a motivation for its development.

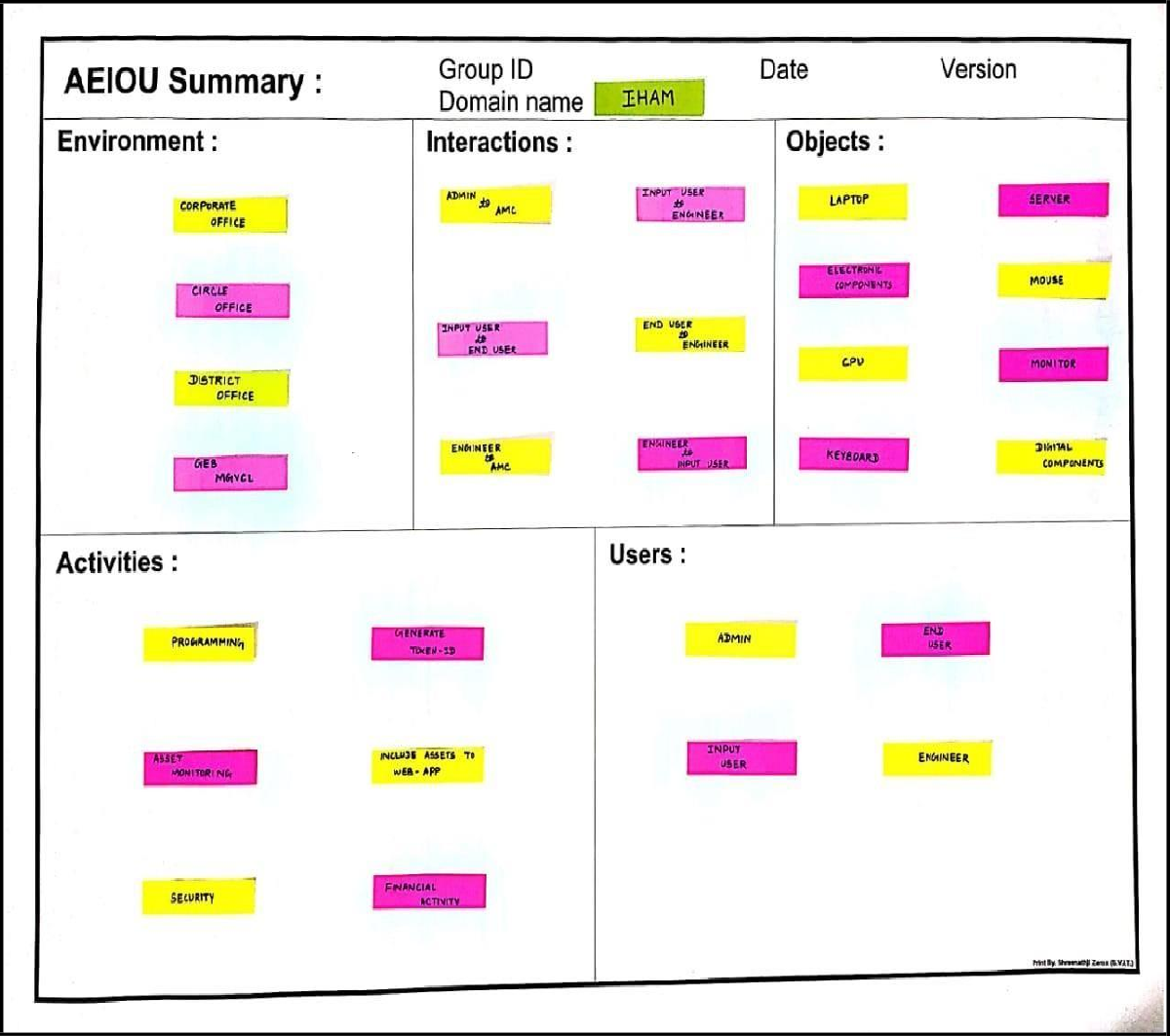
To develop this project we make use of latest DotNet (MVC) technology, which gave us beneficiary and ease of coding environment and also we learnt to manage ERP( enterprise research planning- a software to manage business data) , we also give best services to the employers through our web-application

Now as all are aware about this web-app so employees can use this app for further valuable assets monitoring and can make updates as per their requirement in this web-application.

**Chapter 7: Canvases and Modules**

# AEIOU Summary

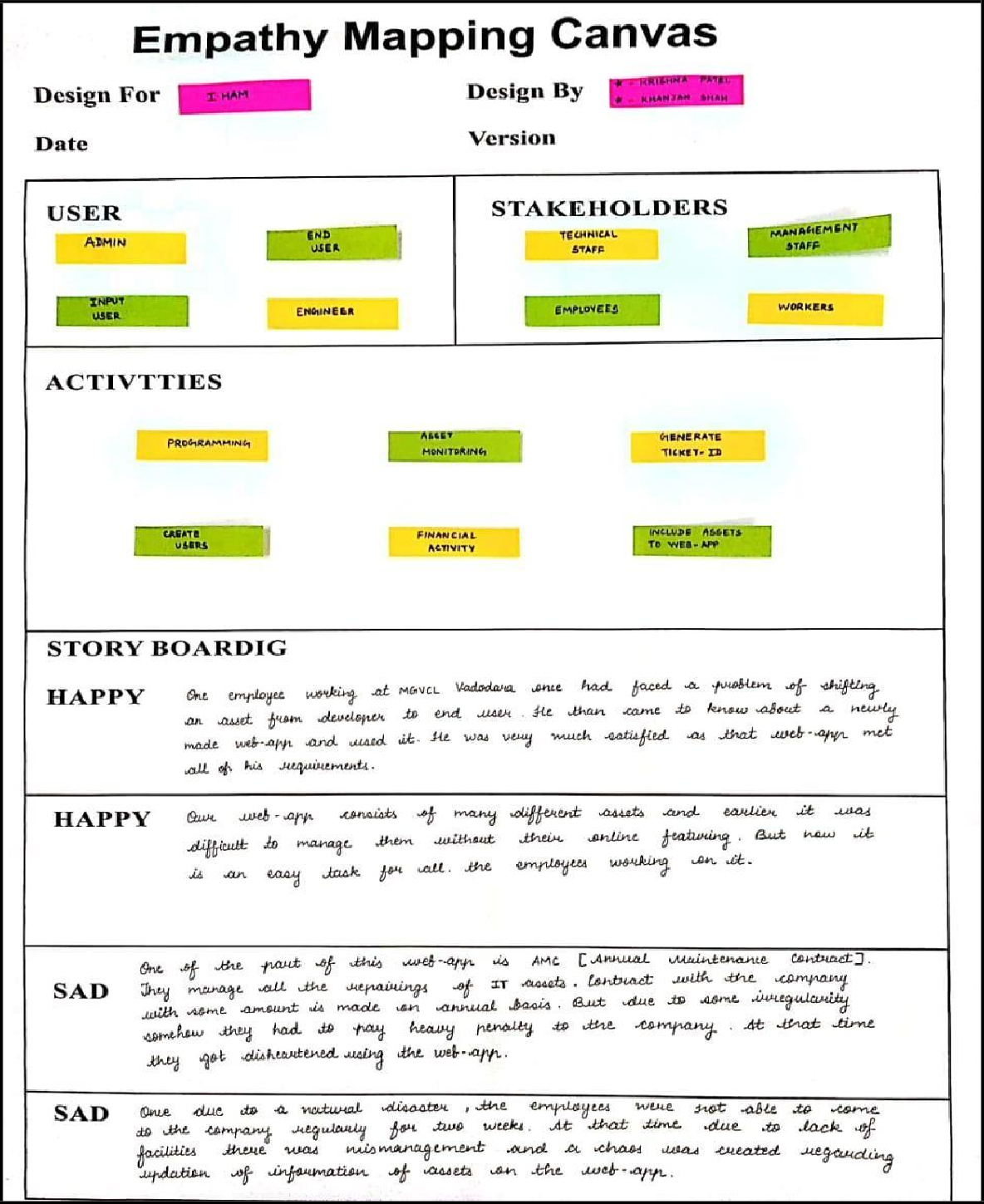
This canvas deals with the environment, activity, objects, users and interactions done at that particular place.



**Fig.7.1 AEIOU Canvas**

# Empathy Mapping Canvas

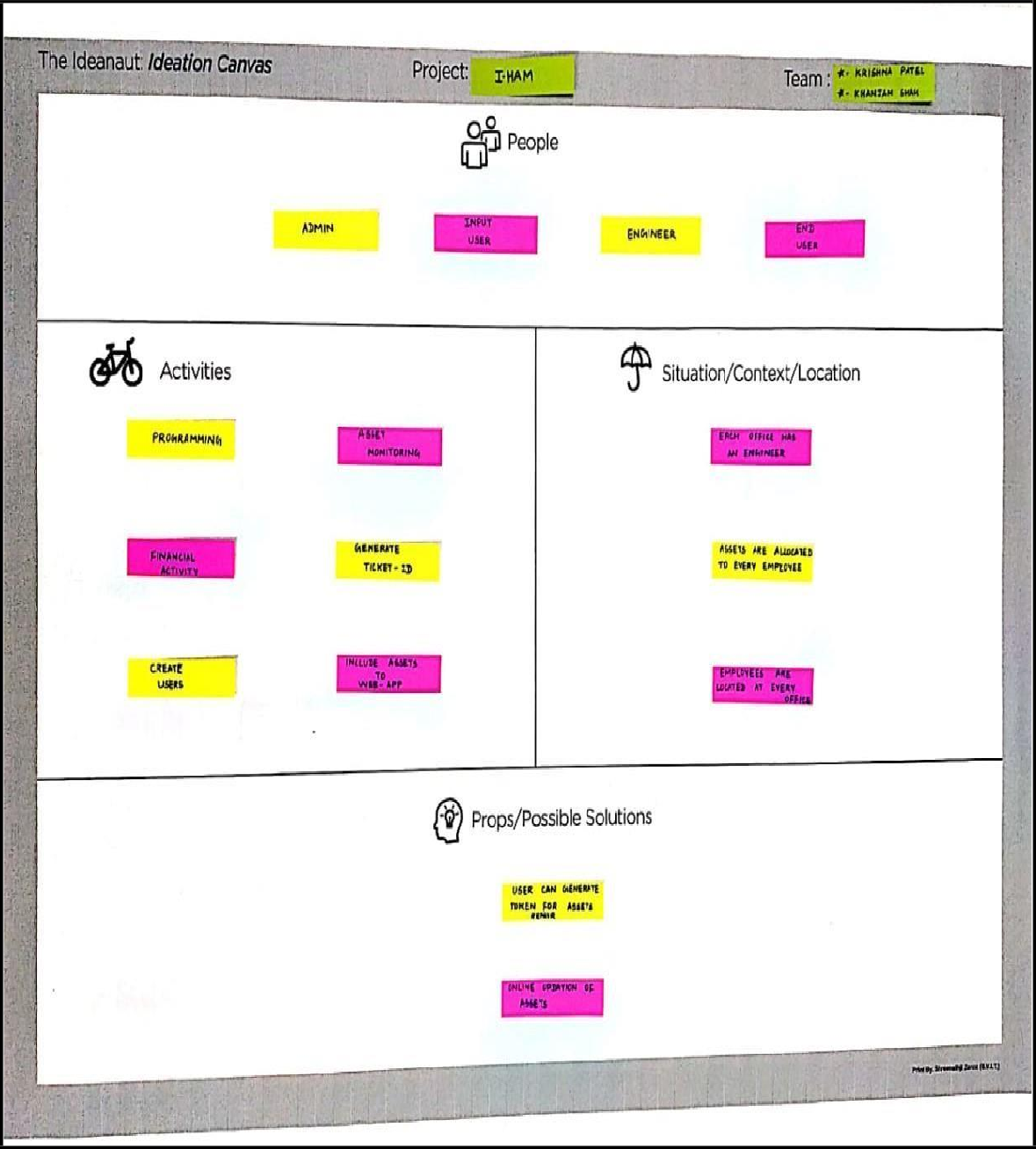
The empathy canvas generally tell us about the positives and the negatives that occurred by considering this project.



**Fig.7.2 Empathy Canvas**

# Ideation Canvas:

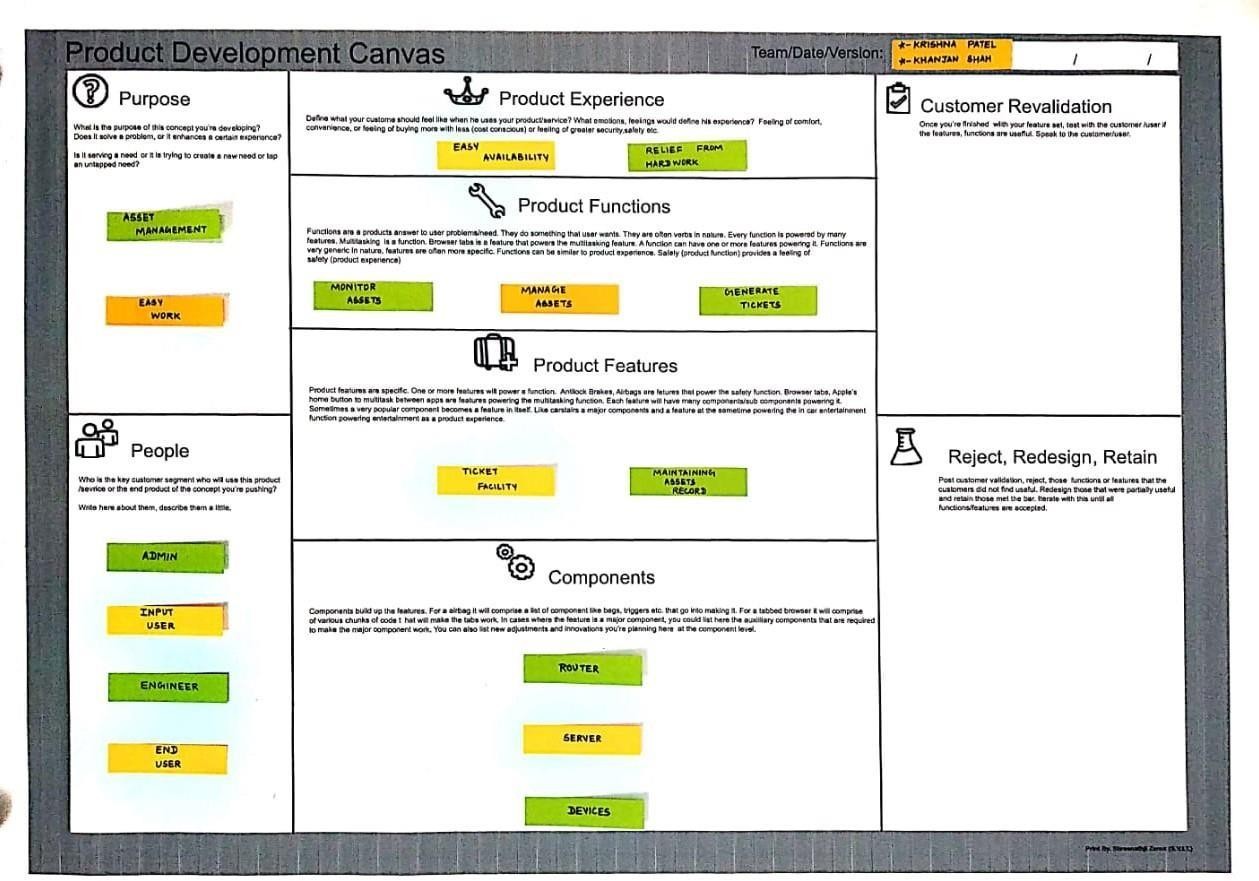
The ideation canvas helps us to generate an idea about the people, activities, and possible solutions to the project.



**Fig.7.3 Ideation Canvas**

# Product development Canvas:

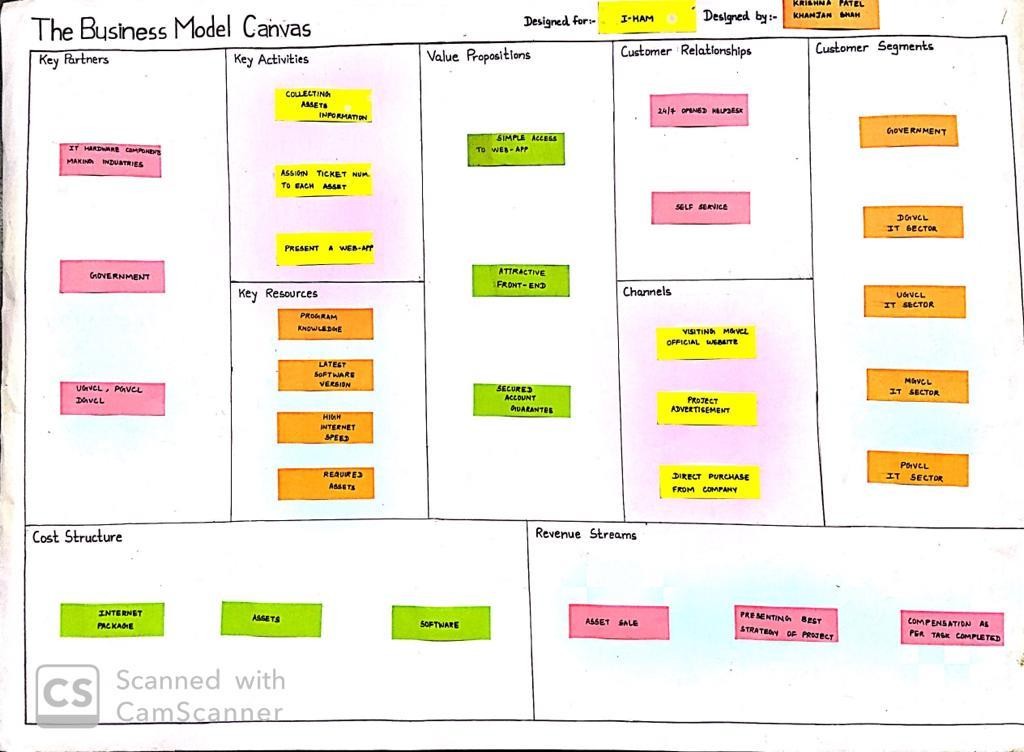
The product development canvas helps to generate the initial thought of the final product that is to be created by its functions, purpose, features etc.



**Fig.7.4 Product development**

# Business Model Canvas :

Business model canvas is the record of our business assets like: our key partners, activities, customer relationship, resources that we used, channels and its cost structure.



# Fig.7.5 Business Model Canvas

**Chapter 8: References** [www.mgvcl.com](http://www.mgvcl.com/) [www.google.co.in/patent-citations(16)](http://www.google.co.in/patent-citations(16))

<http://patents.google.co.in/patent/US8484150B2> <http://patents.google.com/patent/US7747738B2> <http://patents.google.com/patent/US20140288996A1> <https://patents.google.com/patent/CA2395498C>