INTEGRATION OF LICENSING TOOLS WITH LICENSE MONITORING SYSTEM

A PROJECT REPORT

Submitted by

SUCHITA PATEL (110410107007)
MARGI PATEL (110410107037)

In the fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

Computer Engineering

SARDAR VALLABHABHAI INSTITUTE OF TECHNOLOGY, VASAD

Gujarat Technological University, Ahmedabad

December, 2014

SARDAR VALLABHABHAI INSTITUTE OF TECHNOLOGY

Computer Engineering

2014

CERTIFICATE

Date: 5-12-2014

This is to certify that the project entitled **"INTEGRATION OF LICENSING TOOLS WITH LICENSE MONITORING SYSTEM"** has been carried out by SUCHITA PATEL (110410107007) and MARGI PATEL (110410107037) under my guidance in fulfillment of the degree of Bachelor of Engineering in Computer Engineering (7th Semester) of Gujarat Technological University, Ahmadabad during the academic year 2014-2015.

Internal guide: **Head of the Department**

Milin Patel   Mr. Bijal Talati
Asst. Professor,   Computer Engineerung
Computer Department ,   SVIT, Vasad
S.V.I.T., Vasad.

<u>AKNOWLEDGEMENT</u>

It gives us a great pleasure and satisfaction in presenting this project report. We would like to take this opportunity to express our sincere gratitude to several people without whose help and encouragement it would have been impossible for us to carry out the desired work.

It is matter of great pleasure for us to submit this project report on **"*Integration of licensing tools with license monitoring system"*,** as a part of curriculum of **"Bachelor Engineering (C.E)".**

First of all we would like to thank L&TChiyoda for giving us the opportunity to pursue this project in the organization and our guide **Mr. Samirkumar J Patel**, IT Manager for his consistent guidance throughout the project.

We are also very thankful to our internal guide Asst. Prof. **Milin Patel** for their constant encouragement and able guidance.

We are grateful to Mrs**. Bijal Talati,** HOD of Computer Engineering Department, SVIT, Vasad who supported us for accomplishing this project.

At the last but not least, we would like to express our profound gratitude to our parents, who had encouraged& inspired us with their cooperation and their blessings.

ABSTRACT

A Company uses many different engineering software to perform various engineering activity. This software are having network based licenses. The key information of the licensed software need to be maintained in order to make them cost effective and usage oriented. The license key are requested based on the project requirement and hence, requested after various approvals to the software company. After receipt of the license key, installation of the same is required to be done on timely basis so as to allow timely use of the same.

The project is to develop for automating the process of license management with installation and reflecting the same by integrating with the in-house system of license monitoring. In the system, the licenses will be automatically installed and the system will be updated accordingly. The information regarding the previously installed licenses and newly installed licenses will be reported via mail to the designated person. The license mechanism for installation of products of M/s Intergraph and M/s Aveva products will be automated.

The platform used for this will be .NET and back-end database as Oracle.

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

Symbol Name	Abbreviations and symbols
Use case	
Actor	
System boundary	
Class notation	
Entity	
Process	
Data store	
Dataflow	
Decision symbol	
Fork transition	
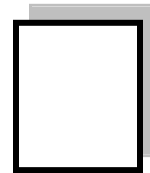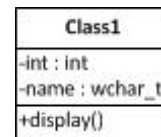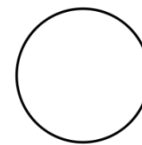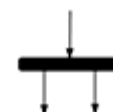
TABLE OF CONTENTS

CHAPTER 1:

COMPANY PROFILE

ABOUT THE COMPANY

Company Name: L&T-CHIYODA LIMITED.

L&T-CHIYODA LIMITED (LTC) is an engineering consultancy organization formed by Larsen & Toubro Limited, India's premier engineering, manufacturing and construction Company (holding 50% equity) & Chiyoda Corporation, Japan, world renowned Engineering Company with five decades of experience in Hydrocarbon and related fields (holding 50% equity).

Incorporated on 19th November 1994, LTC commenced operations in February 1995 and is catering to national & international clients, both directly and through its parent companies. LTC offers international grade engineering and project management services with integrated engineering concepts, supported by state-of-the-art computer hardware and software facilities operating in a networking environment.

LTC, the youngest organization of its kind to get the ISO 9001 certification, has established an independent identity amongst major clients and process know-how suppliers globally, through its indigenous and export engineering credentials. It has upgraded its ISO certification to ISO 9001:2008 and also achieved certifications for ISO 14001:2004, ISO 27001:2005 and OHSAS 18001:2007.

Working towards positive engineering through plant modeling in electronic media, LTC offers a creative response to client's needs. The actual plant itself is 'visualized' to a very close reality in the engineering office during the detailed engineering stage, resulting in high efficiency and accuracy in engineering and ease of construction

LTC has specialized in the engineering for fast track EPC jobs of multiple complexities; repeatedly proving it's adaptability from time to time.

The major industries in which LTC adds significant dimension includes Petroleum refining, Petrochemicals, Chemicals, Fertilizers, Oil & Gas and LNG & LPG.

CHAPTER 2:

INTRODUCTION TO PROJECT

A software license is a legal instrument governing the use or redistribution of software. A typical software license grants an end-user permission to use one or more copies of software in ways. In addition to granting rights and imposing restrictions on the use of software, software licenses typically contain provisions which allocate liability and responsibility between the parties entering into the license agreement.

Software licenses can generally be fit into the following categories: proprietary licenses and free and open source.

Proprietary software licenses
The hallmark of proprietary software licenses is that the software publisher grants the use of one or more copies of software under the end-user license agreement(EULA), but ownership of those copies remains with the software publisher (hence use of the term "proprietary"). This feature of proprietary software licenses means that certain rights regarding the software are reserved by the software publisher. Therefore, it is typical of EULAs to include terms which define the uses of the software, such as the number of installations allowed or the terms of distribution.

The most significant effect of this form of licensing is that, if ownership of the software remains with the software publisher, then the end-user *must* accept the software license. In other words, without acceptance of the license, the end-user may not use the software at all. One example of such a proprietary software license is the license for Microsoft Windows. Software licensing also includes maintenance.

Free and open source software licenses
Free and open-source licenses generally fall under two categories: Those with the aim to have minimal requirements about how the software can be redistributed (permissive licenses), and those that aim to preserve the freedoms that are given to the users by ensuring that all subsequent users receive those rights.

2.1 Current scenario

Today the company requests the licensed software through mail to the vendors according to the requirement of their projects. The vendors provide the proprietary license in response to the mail of request. These licenses can be installed by providing the license key or the file to the license managing tool of the respective vendor. Even though the license managing tools install the licenses for the clients but it also requires a lot of manual work. The software administrator need to provide the key/file as an input to the tool, deal with the tool, keep track of the progress of the process and on installation, updating the License Monitoring System(LMS) accordingly. Due to some reasons, the licensing tool also fails to install the licenses for which no error is also displayed to the client which leads the client to go through the whole process again and check out the reason of the failure.

2.2 Proposed system

The proposed system will allow the automatic installation of the licenses and provide the report to the software administrator via mail. These reduce the manual work up to great extends, updates the LMS system accordingly and also reflect the messages of the process completion and errors, if any.

The administrator will provide the key/file to the tool as an input and the remaining entire process will be done the system itself. On getting the input, the system will check for the conditions that need to be satisfied for the key installation process and if it does not match, keeps continuous check for the condition and as soon as it meets the condition, it proceed for the installation. It will generate a report at the end of the process and send it to the administrator via mail. Also the LMS system will be updated accordingly by the system.

CHAPTER 3:

SYSTEM ANALYSIS

3.1 Problem Statement

The licensing tool installs the license with the overhead of manual work. Also the absence of the software license administrator there are problems that are arising. The tool may fail to install the license and even do not acknowledge with the error message. The LMS system which displays the distribution of licenses is to be updated after latest installation process manually.

Integrating the licensing tool and the LMS system will reduce the lot of manual work load and even automatically updates the changes and reflects them on the system. On any error, it will also give out the error message relative to the error which was earlier time consuming to find.

3.2 Scope

The license installation procedure is associated with many constraints. The system will allow the license software administrator to install the license with ease and efficiently. This will provide a cost and time effective solution.

The scope of the project is well defined which is as follows:

Design and develop the system which includes
- Automatic License Installation and Activation
- Generation of the report on installation
- Making the changes on the LMS
- Display of understandable error messages, if any

3.3 Feasibility study

Resource feasibility

Licensing tools like Flex-LM and SPLM are required to carry out the project. Also the existing LMS system, which needs to be updated, is already an in-house developed system whose access can be achieved.

Economic feasibility

This system will not require any cost of developing, as it will be developed with the existing tools and technology which are freely and easily available.

Technical feasibility

The measure of the practicality of a specific solution and availability of technical resources define technical feasibility.

The solution decided for the proposed system is technically possible but there also exist some constraints which may be deal out manually and can be solved in future.

Operational feasibility

Our system will help the software administrator to reduce the overhead of manual work and continuous make a watch on the progress of the process. It will also update the LMS system and reflect the changes on the LMS system which will help to view the availability of the resources. Time efficiency and availability of the system are the good factors provided by the system.

3.4 Model

Waterfall Model

Waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing downwards like a waterfall through the phases of Conception, Initiation, Analysis, Design, Construction Testing, Implementation and Maintenance. It is also known as classic life cycle model.

It suggests a systematic sequential approach for software development. It begins with customer specification of requirements and processes through planning, modeling, construction and deployment.
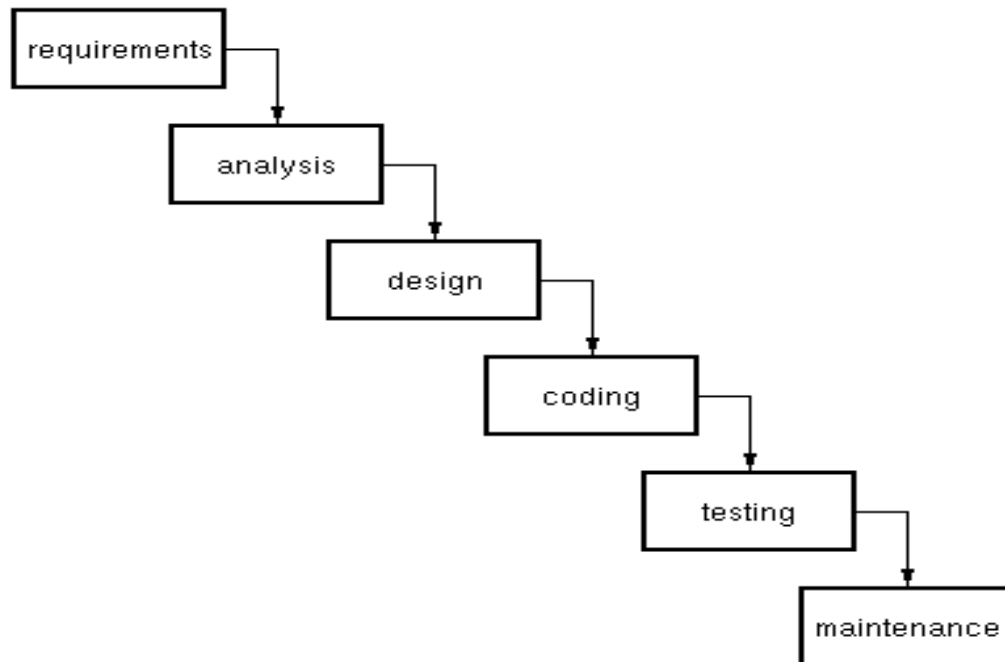

Figure 3.4.1 *Waterfall Model*

Waterfall model is used in the following situations:
- When requirements are very well known, clear and fixed.
- When product definition is stable.
- When technology is understood.
- When project is short.
- When ample resources with expertise are available.

The waterfall model allows departmentalization and managerial control. Each phase has a specific deliverable process. It works well for small projects.
As there are no specific and ambiguous requirements this model proves to be favourable for this system.

3.5 Gantt chart

ID	Task Name	Start	Finish	Duration	Q3 14			Q4 14			Q1 15			Q2 15
					Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
1	Project definition	07-07-2014	14-07-2015	267d	██████████████████████████████									
2	Learning licensing tools & LMS	07-07-2014	28-07-2014	16d	█									
3	Current scenario	28-07-2014	11-08-2014	11d		█								
4	Requirement gathering	11-08-2014	05-09-2014	20d		█								
5	System analysis	08-09-2014	29-09-2014	16d			█							
6	System design	30-09-2014	22-10-2014	17d				█						
7	Implementation	27-10-2014	05-03-2015	94d					████████					
8	Testing and debugging	06-03-2015	27-04-2015	37d									██	

Figure 3.5.1 *Gantt chart*

CHAPTER 4:

REQUIREMENT ANALYSIS

4.1 Functional requirements

4.1.1 Install license key

The License Key / License File will be provided by different vendors for different software according to the request made. It will be accepted as an input, checks the condition iteratively whenever required, necessary for installation of the products of Intergraph, internally and completes the installation process.

4.1.2 Generate output files

Intergraph:
   The output file generated will consist of the audit code and
   the key status file.
 Aveva:
    The output file generated in the case of Aveva is the key status
    file.

4.1.3 Send mail

The information about the final results will be stored in the file which will be mailed to the Designated Manager. Hence the manager can know about the process done successfully and there is a record available with him for reference.

4.1.4 Update LMS

According to the currently installed Licenses, respective updations are done in License Monitoring System (LMS).

4.1.5 Other functionality

Manage Server List
Although the servers are fixed for the licenses, there might be situations that will demand to change the server, reconfigure it or add new servers or remove some of them. Managing the servers whenever required will also be facilitated.

Daily track of Activation Dates
There will be a daily watch on the activation dates of the Licenses and if required, a call to activate the License on the day of activation will be made.

Display Error Messages
In case if the license is not installed properly on the occurrence of an error then instead of searching for them manually, error will be displayed on the user screen.

4.2 Non functional requirements

4.2.1 Availability

The system helps to improve the availability of the software to the employees of the organization and the administrator with reliable data, as it will timely update the LMS System. Also it can be accessed from any machine in the organization network. So it leads to accuracy in terms of the available and software-in-use records on LMS.

4.2.2 Security

As the licenses are very sensitive to handle, only authorized and equipped person should able to deal with it. Hence the login and password facility will be provided to ensure that the only authorized persons are given the rights to access the system for installation and can make modifications.

4.2.3 Reliability

The system will ensure that reliable data is placed on the LMS data store and the same data will be reflected on the screen.

4.3 System requirements

Software specifications for development

Platform: .NET framework
Technology: ASP.net
Language: VB.net
Back End: Oracle
For development: Visual Studio .Net Tools
For design: Visual Studio Designer

Software specifications for deployment

Operating system: Windows NT and above
Server: IIS (Internet Information service): configured
Back End: Oracle

Hardware Interface for development
Processor: i3
RAM: 1 GB or above
Hard Disk: 5GB

Hardware Interface for deployment
Processor: i3
RAM: 1 GB
Hard Disk: 2GB

CHAPTER 5:

SYSTEM DESIGN

5.1 Use cases
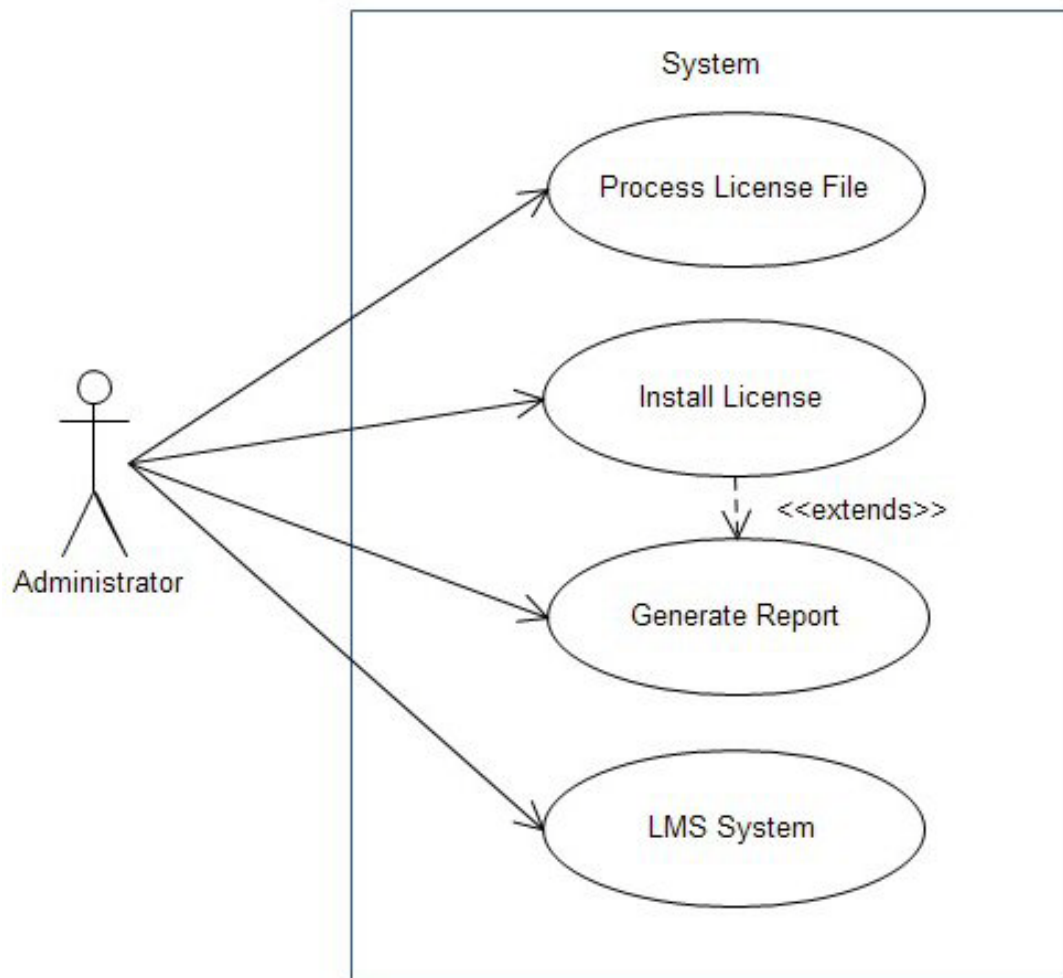
Figure 5.1.1 *Usecase*

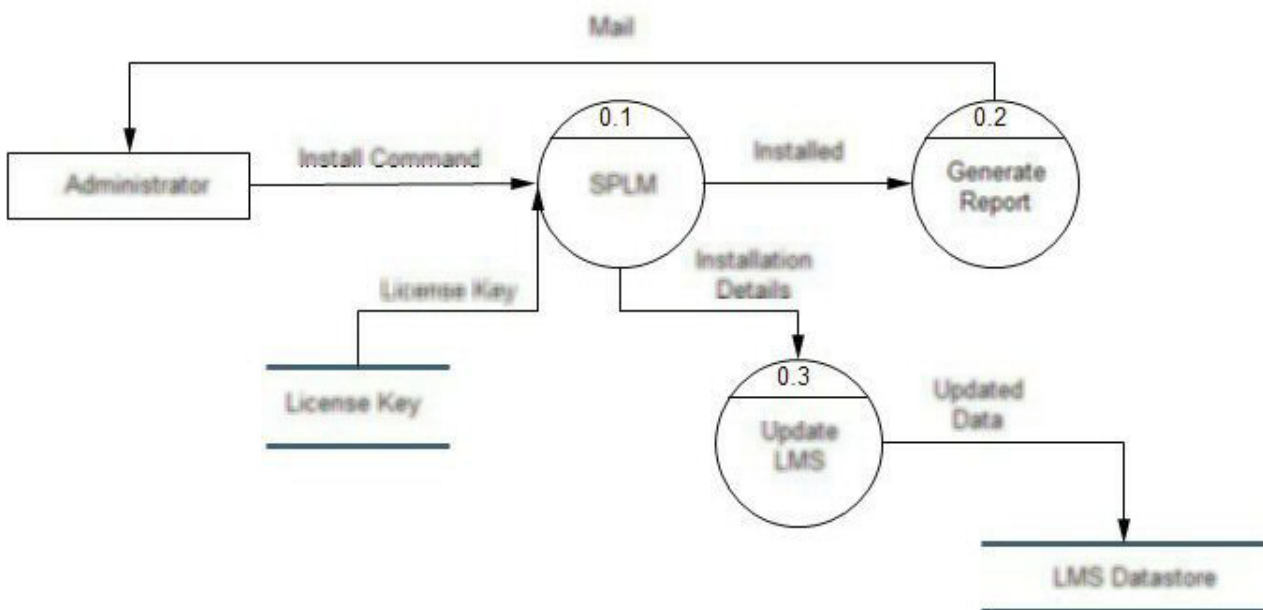**5.2 Class diagram**
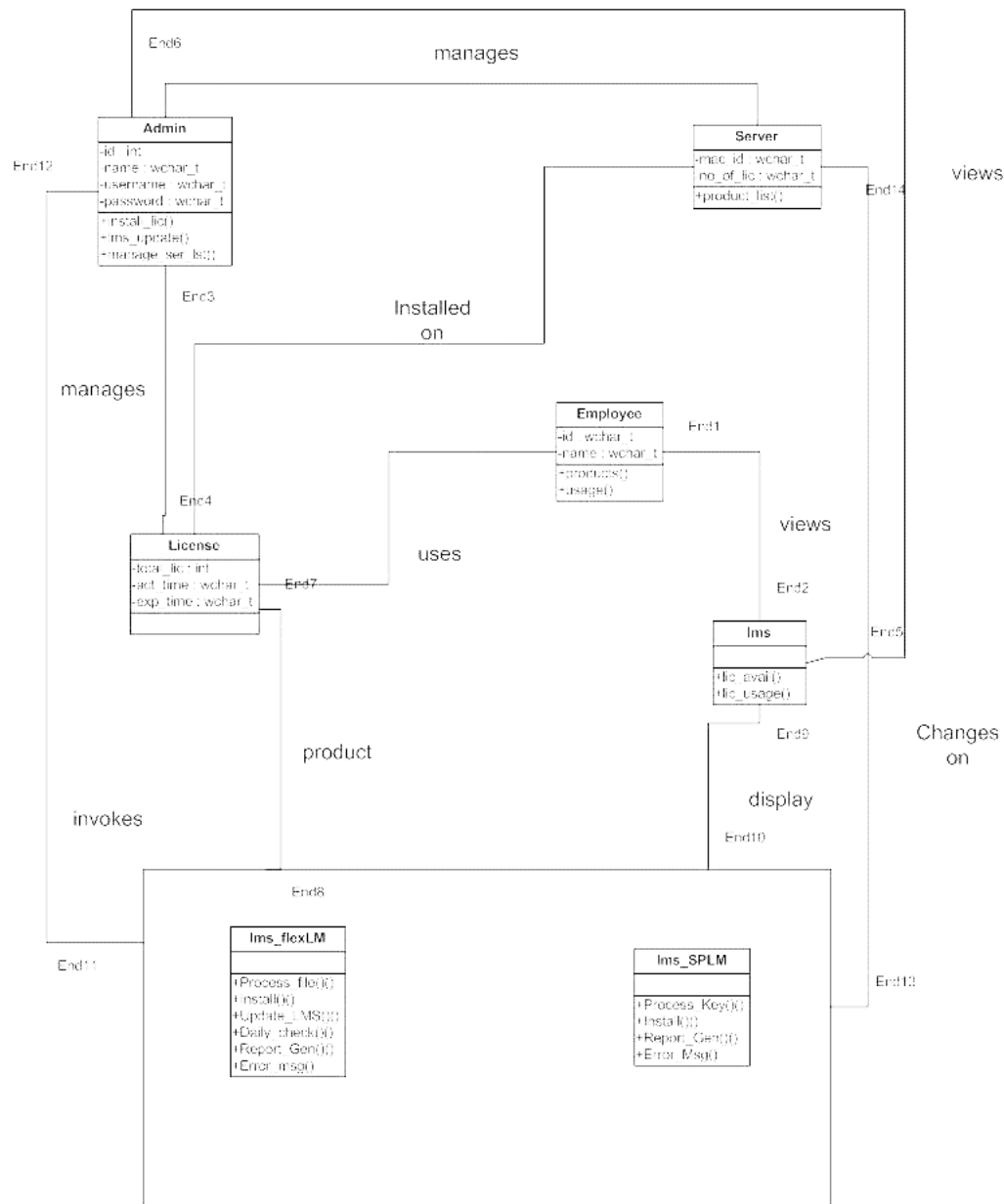
Figure 5.2.1 *Class diagram*

5.3 Data Flow Diagram

Data flow diagram for Intergraph:

DFD level 0:
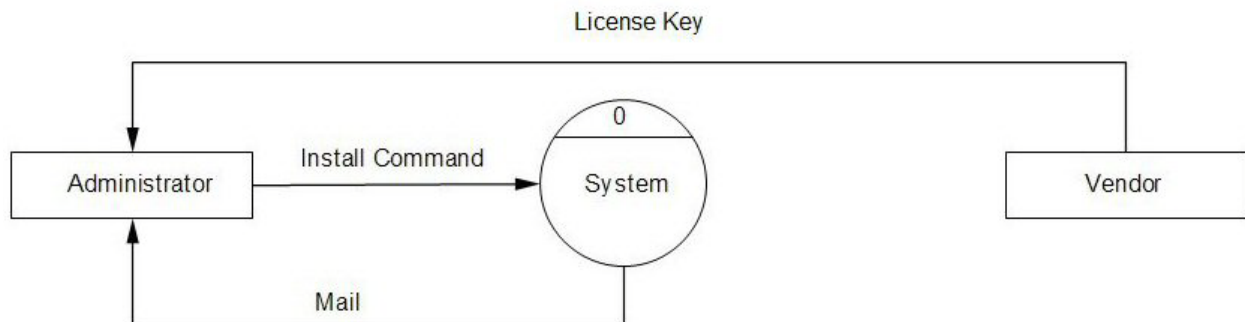
DFD level 1:
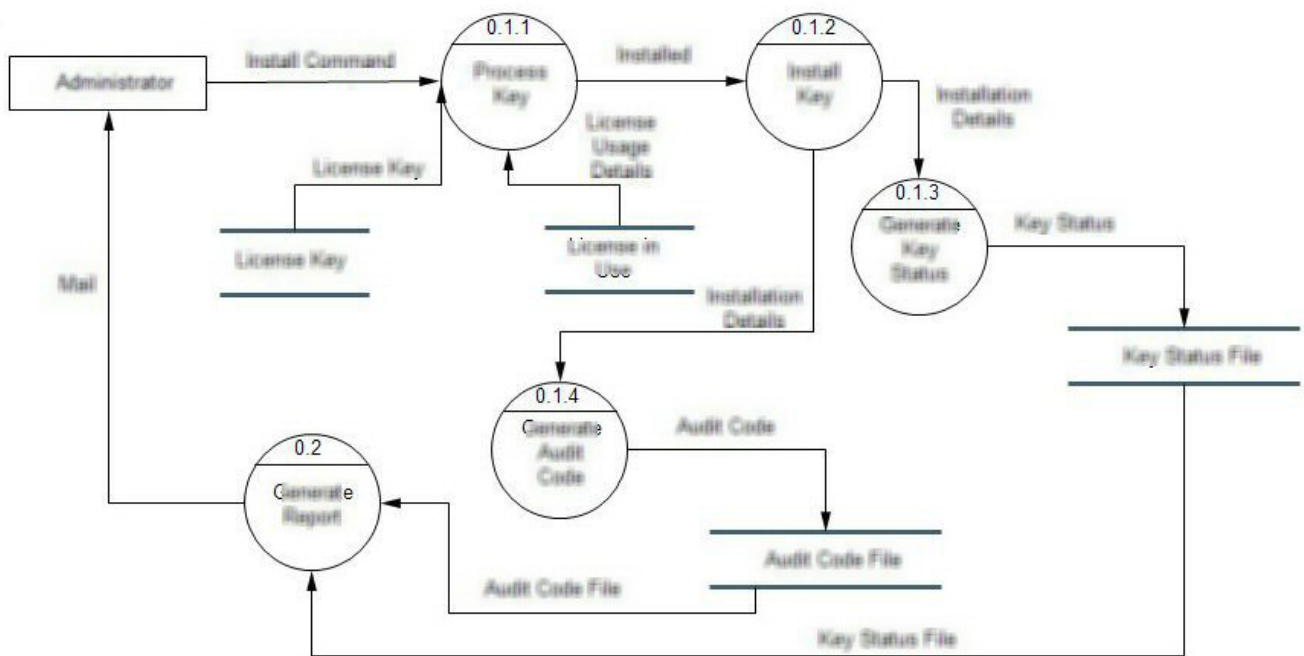
DFD level 2:

Figure 5.3.1 DFD intergraph

Data flow diagram for Aveva:

DFD level 0:
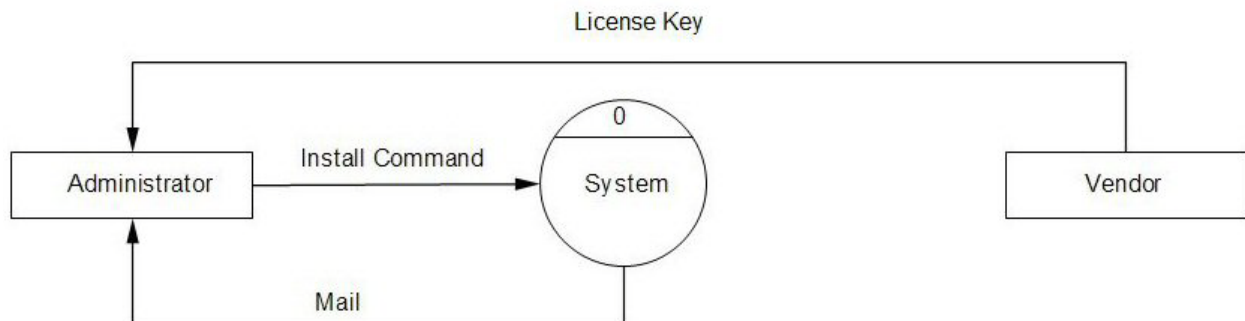
DFD level 1:
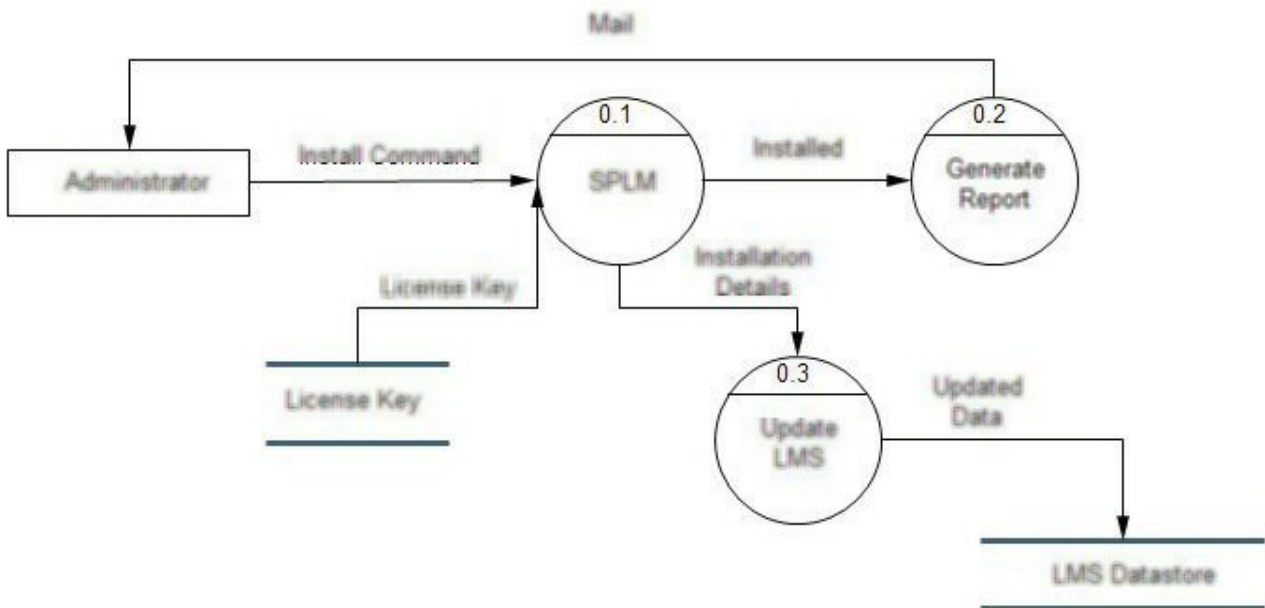
DFD level 2:
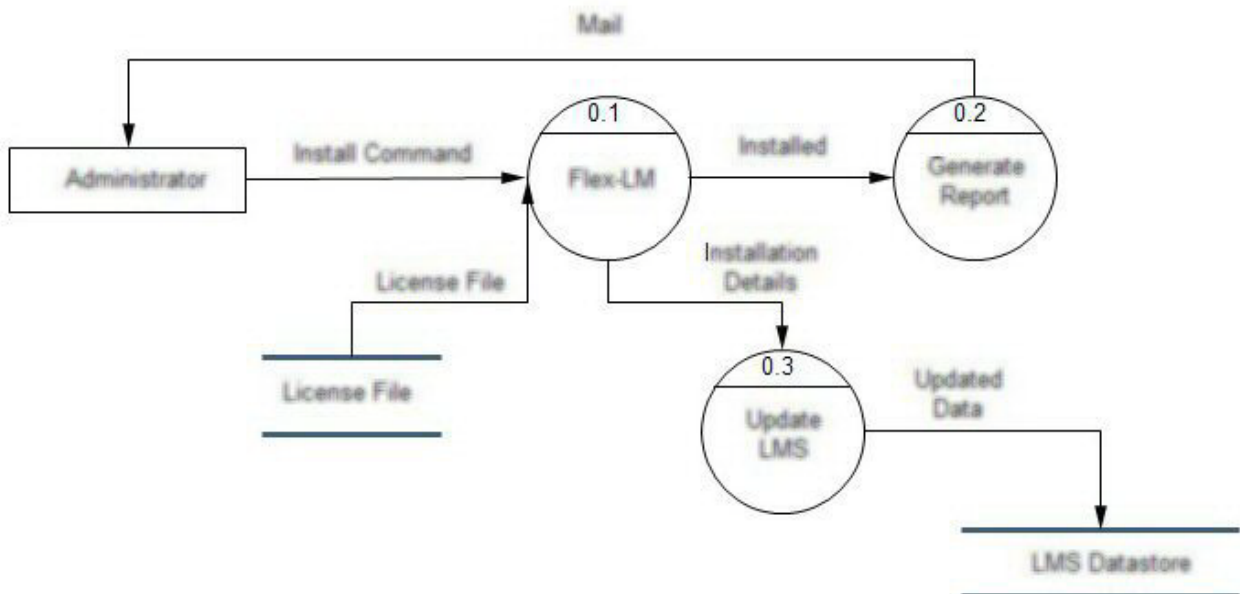
Figure 5.3.2 *DFD aveva*

5.4 State diagram
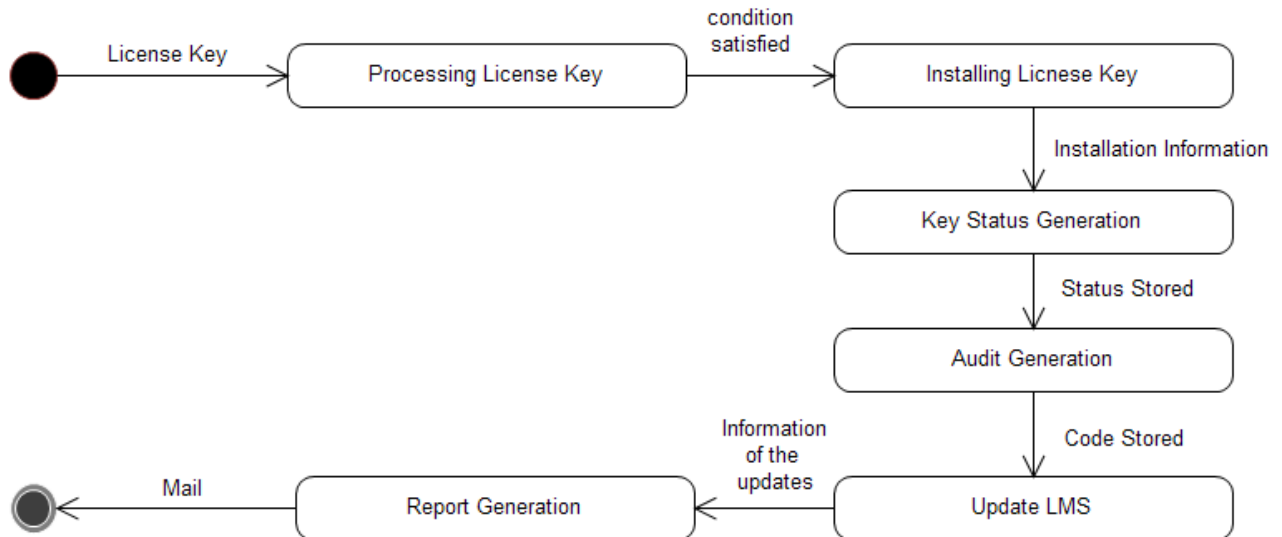
Figure 5.4.1 *State diagram Intergraph*
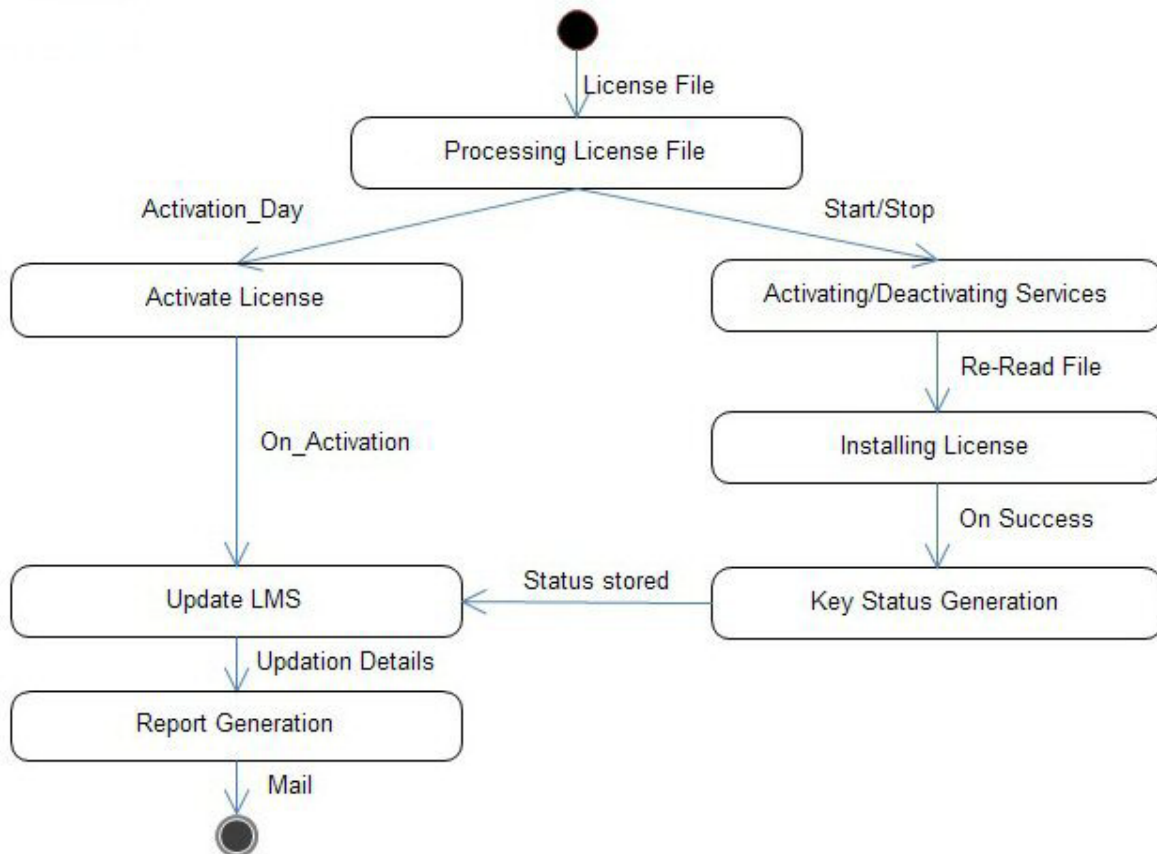
Figure 5.3.2 *State diagram Aveva*

5.5 Activity Diagram
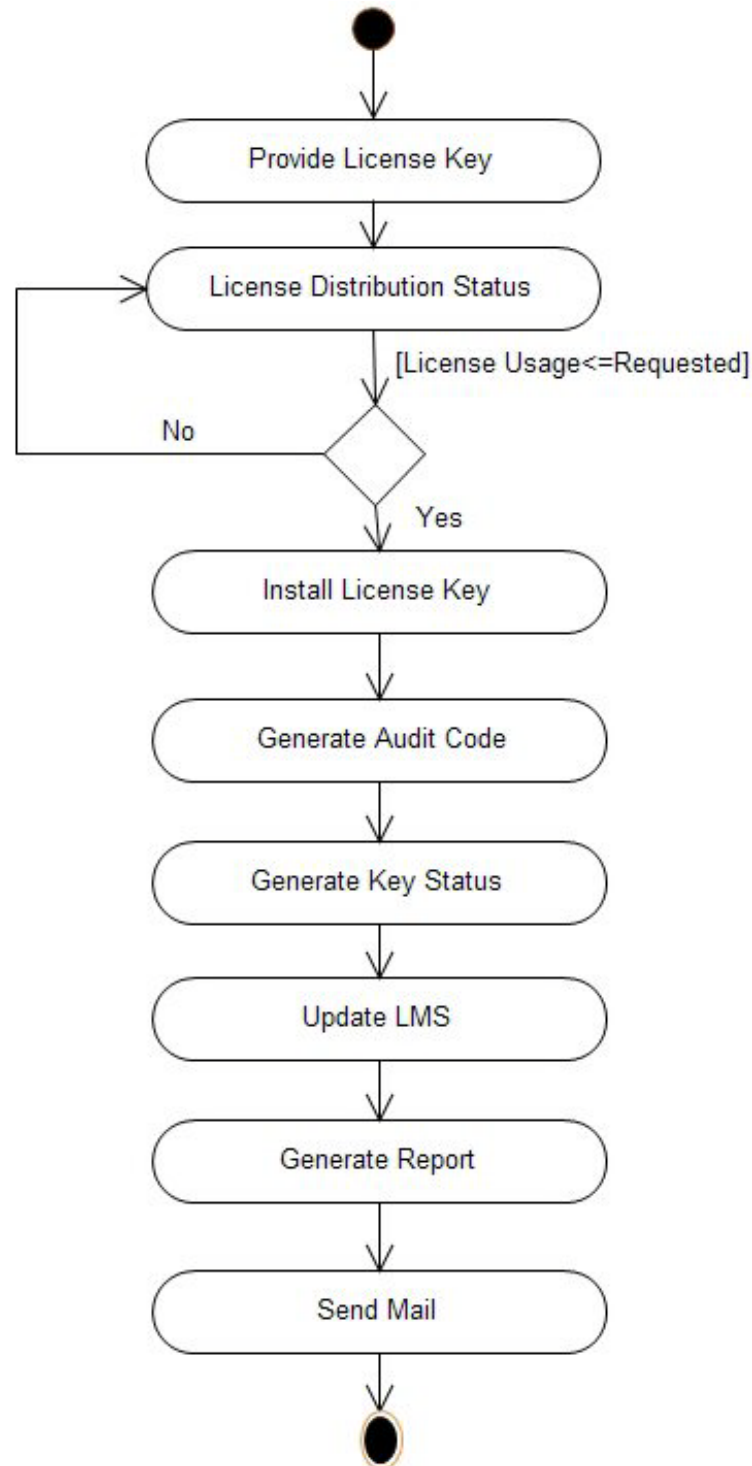
Figure 5.5.1 *Activity diagram Intergraph*
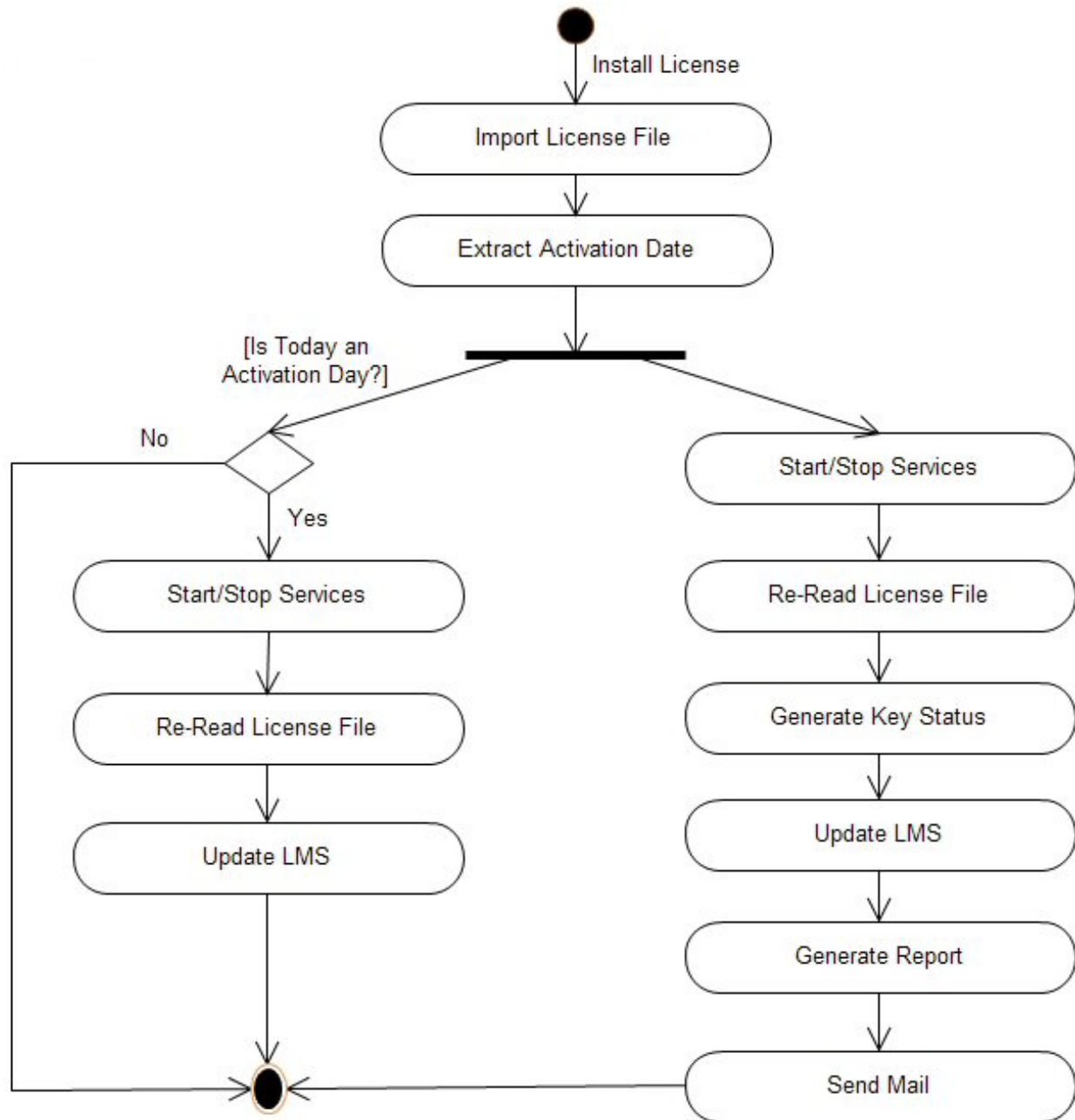
Figure 5.5.2 *Activity diagram Aveva*

CHAPTER 6:

IMPLEMENTATION

6.1 Implementation methodology

6.1.1 Observation Matrix

6.1.2   Ideation Canvas
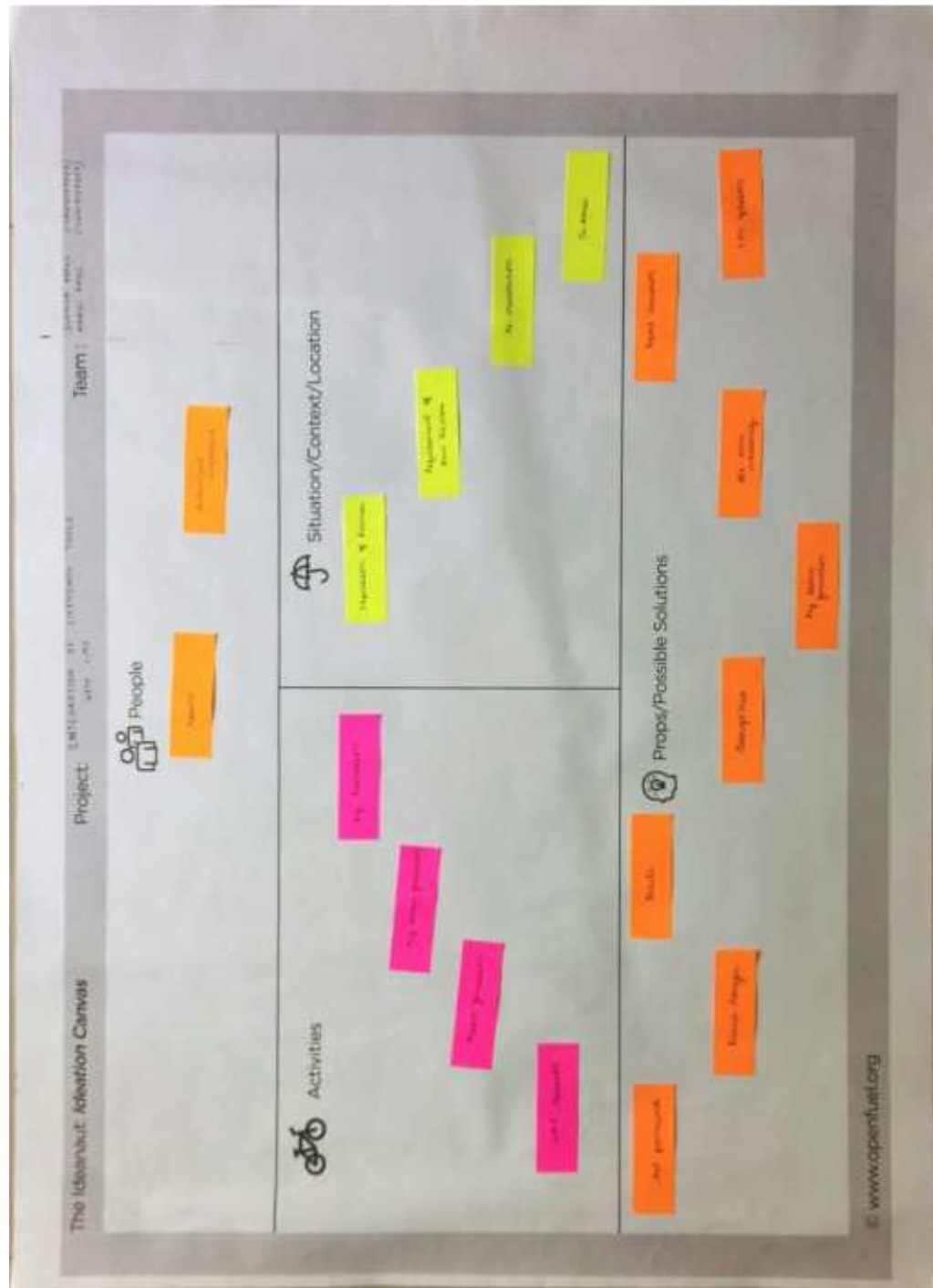
6.2 Implementation

Screenshots of the work done are shown below

Login page: Here the user will need to enter the id given and password to go for installation.

Figure 6.1 *Login page*

Installation for Intergraph: The licenses for the products of intergraph are installed here, where the server is selected and license key is given as input.

Figure 6.2 *Installation for SPLM*

Installation for aveva: The licenses for the products of aveva are installed here, where the server needs to be selected and license file is given as input.

Figure 6.3 *Installation for FlexLM*

CHAPTER 7:
REFERENCES

Books: ASP.net 4 in c# and vb.net by Scott Hanselman, Devin Rader, Bill Evje.
Black Book of .Net programming
Black Book of VB.Net

Websites:
www.CodeProject.com
www.Google.com
www.tutorialspoint.com
msdn.microsoft.com