GEDİZÜNİVERSİTESİ
izmir

Department of Computer Engineering

COM 401–SOFTWARE ENGINEERING

PROJECT PROPOSAL

Academic Year: Fall 2015-2016

Course Instructors: Prof.Dr.Haluk Gümüşkaya

Course Assistant: Ins.Gokhan Akyol

CUSTOMER TRACKING SYSTEM

18.10.2015

CONTENTS

,

1.Introduction

This is the customer tracking system of AN software company. The system is made to make sales of web services online(for example hosting service, web design etc.).

Because the firm is making their sales with large sums,sales are made by a contract like a prepared offer which is made according to the customers requirements. So the sale is completely made using the system. System keeps the record of all sales in a database, so in long term the sales manager can prepare offers according to same customers previous sales and other factors.

Until now all data and contracts were held in folders. Thanks to the system all contracts are held on a database and the manager is able to see previous contracts for a person and more, so he can decide his offers according to previous sales. And also in long term he can see statistics. This system enables the manager to see the statistics and have some idea about the rate of profit.

1.1.Purpose of the Project

The purpose of this project is until now all data and contracts were held in folders. Thanks to the system all contracts are held on a database and the manager is able to see previous contracts for a person and more, so he can decide his offers according to previous sales. And also in long term he can see statistics. This system enables the manager to see the statistics and have some idea about the rate of profit.

1.2.Project Members, Scope and Outcomes

The scope of the Project is to hold every invoices that are made. So that, employees and the management reaches any of them any time they need easily. So, they can have information about the customers, the invoices and they can evaluate their situation and decide about their future sales.

As an outcome, the profitability of the firm is highly increased due to the strategies of the firm which is regulated according to the needs. Those factor that effects the strategies are; the amount of purchases of a single customer will cause discount on sales, the total price of the sale will cause discount on the sale too.

Project Members

E.Berkan SÖNMEZ
Gediz University
Department: Computer Engineering
E-mail: berkan.sonmez@gmail.com
Role(s) : Developer,Tester
Ozan YALDIR
Gediz University
Department: Computer Engineering
E-mail: ozanyaldir@gmail.com
Role(s) : Developer, Tester
İlham GÜLTEKİN
Gediz University
Department: Computer Engineering
E-mail: ilhamgultekin@gmail.com
Role(s) : Scrum Master
Eray BAYRAM
Gediz University
Department: Computer Engineering
E-mail: eraybayram.sony@windowslive.com
Role(s) : Product Owner

2.Project Details

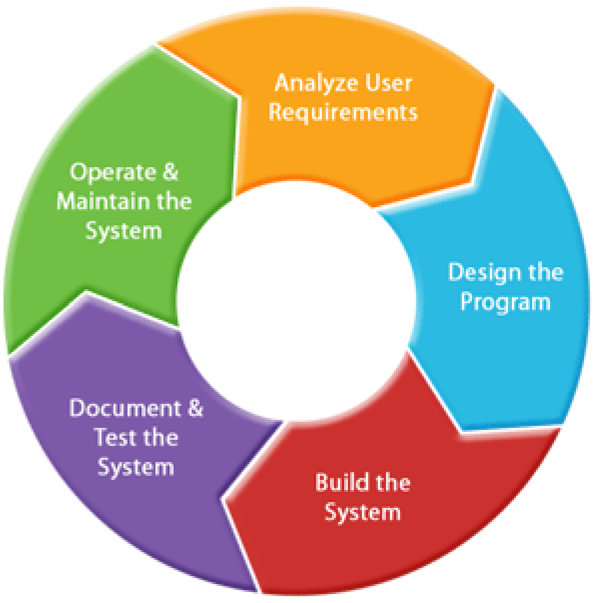
Analyze User Requirements
Design the Program
Build the System
Document & Test the System
Operate & Maintain the System

Figure 2.1: Software Project Life Cycle

- **PHP Laravel Framework**

Laravel is a free, open-source PHP web application framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance, and its orientation toward syntactic sugar. As of March 2015, Laravel is regarded as one of the most popular PHP frameworks, together with Symfony2, Nette, CodeIgniter, Yii2 and other frameworks. The source code of Laravel is hosted on GitHub and licensed under the terms of MIT License.

Taylor Otwell created Laravel as an attempt to provide a more advanced alternative to the CodeIgniter framework, which did not provide certain features such as built-in support for user authentication and authorization. Laravel's first beta release was made available on June 9, 2011, followed by the Laravel 1 release later in the same month. Laravel 1 includes built-in support for authentication, localisation, models, views, sessions, routing and other mechanisms, but lacks support for controllers that prevents it from being a true MVC framework.

Laravel 2 was released in September 2011, bringing various improvements from the author and community. Major new features include the support for controllers, which made Laravel 2 a fully MVC-compliant framework, built-in support for the inversion of control (IoC) principle, and a templating system called Blade. As a downside, support for third-party packages was removed in Laravel 2.

Laravel 3 was released in February 2012 with a set of new features including the command-line interface (CLI) named Artisan, built-in support for more database management systems, database migrations as a form of version control for database layouts, support for handling events, and a packaging system called Bundles. An increase of the Laravel's userbase and popularity lined up with the release of Laravel 3.

Laravel 4, codenamed Illuminate, was released in May 2013. It was made as a complete rewrite of the Laravel framework, migrating its layout into a set of separate packages distributed through Composer, which serves as an application-level package manager. Such a layout improved the extendibility of Laravel 4, which was paired with its official regular release schedule spanning six months between minor point releases. Other new features in the Laravel 4 release include database seeding for the initial population of databases, support for message queues, built-in support for sending different types of email, and support for delayed deletion of database records called soft deletion.

Laravel 5 was released in February 2015 as a result of internal changes that ended up in renumbering the then-future Laravel 4.3 release. New features in the Laravel 5 release include support for scheduling periodically executed tasks through a package called Scheduler, an abstraction layer called Flysystem that allows remote storage to be used in the same way as local file systems, improved handling of package assets through Elixir, and simplified externally handled authentication through the optional Socialite package. Laravel 5 also introduced a new internal directory tree structure for developed applications.

Laravel 5.1, released in June 2015, is the first release of Laravel to receive long-term support (LTS), with planned availability of bug fixes for two years and security patches for three years. LTS releases of Laravel are planned to be released every two years.

The following features serve as Laravel's key design points (where not specifically noted, descriptions refer to the features of Laravel 3)

- Bundles provide a modular packaging system since the release of Laravel 3, with bundled features already available for easy addition to applications. Furthermore, Laravel 4 uses Composer as a dependency manager to add framework-agnostic and Laravel-specific PHP packages available from the Packagist repository.
- Eloquent ORM (object-relational mapping) is an advanced PHP implementation of the active record pattern, providing at the same time internal methods for enforcing constraints on the relationships between database objects. Following the active record pattern, Eloquent ORM presents database tables as classes, with their object instances tied to single table rows.
- Query builder, available since Laravel 4, provides a more direct database access alternative to the Eloquent ORM. Instead of requiring SQL queries to be written directly, Laravel's query builder provides a set of classes and methods capable of building queries programmatically. It also allows selectable caching of the results of executed queries.
- Application logic is an integral part of developed applications, implemented either by using controllers or as part of the route declarations. The syntax used to define application logic is similar to the one used by Sinatra framework.
- Reverse routing defines a relationship between the links and routes, making it possible for later changes to routes to be automatically propagated into relevant links. When the links are created by using names of existing routes, the appropriate uniform resource identifiers (URIs) are automatically created by Laravel.
- Restful controllers provide an optional way for separating the logic behind serving HTTP GET and POST requests.
- Class auto loading provides automated loading of PHP classes without the need for manual maintenance of inclusion paths. On-demand loading prevents inclusion of unnecessary components, so only the actually used components are loaded.

- View composers serve as customizable logical code units that can be executed when a view is loaded.
- Blade templating engine combines one or more templates with a data model to produce resulting views, doing that by transpiling the templates into cached PHP code for improved performance. Blade also provides a set of its own control structures such as conditional statements and loops, which are internally mapped to their PHP counterparts. Furthermore, Laravel services may be called from Blade templates, and the templating engine itself can be extended with custom directives.
- IoC containers make it possible for new objects to be generated by following the inversion of control (IoC) principle, in which the framework calls into the application- or task-specific code, with optional instantiating and referencing of new objects as singletons.
- Migrations provide a version control system for database schemas, making it possible to associate changes in the application's codebase and required changes in the database layout. As a result, this feature simplifies the deployment and updating of Laravel-based applications.
- Database seeding provides a way to populate database tables with selected default data that can be used for application testing or be performed as part of the initial application setup.

MVC(Model View Controller)

Model–view–controller (MVC) is a software architectural pattern for implementing user interfaces. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user.Traditionally used for desktop graphical user interfaces, this architecture has become extremely popular for designing web applications.

MVC was one of the seminal insights in the early development of graphical user interfaces, and one of the first approaches to describe and implement software constructs in terms of their responsibilities.Trygve Reenskaug introduced MVC into Smalltalk-76 while visiting Xerox Parc in the 1970s. In the 1980s, Jim Althoff and others implemented a version of MVC for the Smalltalk-80 class library. It was only later, in a 1988 article in The Journal of Object Technology, that MVC was expressed as a general concept.The MVC pattern has subsequently evolved, giving rise to variants such as HMVC, MVA, MVP, MVVM, and others that adapted MVC to different contexts.

The use of the MVC pattern in web applications exploded in popularity after the introduction of Spring framework for Java. The introduction of the frameworks Rails for Ruby and Django for Python, both of which had a strong emphasis on rapid deployment, increased its popularity outside of the traditional enterprise environment in which MVC has long been popular. MVC web frameworks now hold large market shares relative to non-MVC web toolkits.

- **HTML(HyperText Markup Language)**

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

The language is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page.

HTML can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

In 1980, physicist Tim Berners-Lee, then a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an Internet-based hypertext system. Berners-Lee specified HTML and wrote the browser and server software in late 1990. That year, Berners-Lee and CERN data systems engineer Robert Cailliau collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes from 1990 he listed"some of the many areas in which hypertext is used" and put an encyclopedia first.

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991. It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by SGMLguid, an in-house Standard Generalized Markup Language (SGML)-based documentation format at CERN. Eleven of these elements still exist in HTML 4.

HTML is a markup language that web browsers use to interpret and compose text, images and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents. However, the SGML concept of generalized markup is based on elements (nested annotated ranges with attributes) rather than merely print effects, with also the separation of structure and markup; HTML has been progressively moved in this direction with CSS.

- **CSS(Cascading Style Sheets)**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be bold", leaving pure semantic HTML markup that asserts "this text is a level 1 heading" without formatting code such as a <bold> tag indicating how such text should be displayed.

This separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to display the web page differently depending on the screen size or device on which it is being viewed. Although the author of a web page typically links to a CSS file within the markup file, readers can specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author has specified. If the author or the reader did not link the document to a style sheet, the default style of the browser will be applied. Another advantage of CSS is that aesthetic changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in one file, rather than by a laborious (and thus expensive) process of crawling over every document line by line, changing markup. .

- **SQL(Structured Query Language)**

SQL (Structured Query Language) is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and a data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks." Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, though, most SQL code is not completely portable among different database systems without adjustments.

SQL was initially developed at IBM by Donald D. Chamberlin and Raymond F. Boyce in the early 1970s. This version, initially called SEQUEL (Structured English QUEry Language), was designed to manipulate and retrieve data stored in IBM's original quasi-relational database management system, System R, which a group at IBM San Jose Research Laboratory had developed during the 1970s. The acronym SEQUEL was later changed to SQL because "SEQUEL" was a trademark of the UK-based Hawker Siddeley aircraft company.

In the late 1970s, Relational Software, Inc. (now Oracle Corporation) saw the potential of the concepts described by Codd, Chamberlin, and Boyce, and developed their own SQL-based RDBMS with aspirations of selling it to the U.S. Navy, Central Intelligence Agency, and other U.S. government agencies. In June 1979, Relational Software, Inc. introduced the first commercially available implementation of SQL, Oracle V2 (Version2) for VAX computers.

After testing SQL at customer test sites to determine the usefulness and practicality of the system, IBM began developing commercial products based on their System R prototype including System/38, SQL/DS, and DB2, which were commercially available in 1979, 1981, and 1983, respectively.

3.Proposed System

3.1. Functional Requirements

•AN Software Sales managers need to login this system with their email and password
•AN Software Yazılım Admin can be login as Admin Role and register new Sales Manager account
•Sales Manager can be registered new Customer
•Customer information : CompanyName,Phone,Fax, Address,TaxNumber, Company Contact Worker name ,email and phone,.
•Sales Manager can be created “sales offer” for customer.
•Sales Offer should be contains : Offer Contract Text, Offer Total Amount, CompanyName and also offer products.
•Sales Offer products should be contains : ProductName, Quantity,UnitPrice,TotalPrice.
•Sales Manager can be added service invoice for approved offers.
•Service Invoice should be contain offer details and customer invoice address.
•Sales Manager can be listed all offers and invoces for any customer.
•Sales Manager can be searched about customer on one screen with Company Contact name,email and company name fields.
•Sales Manager can be gotten pdf .

3.2. Nonfunctional Requirements

Reliability

Only admin and sales manager can connect the system with their personal username and passwords. Other people cannot see customer’s contracts and their personal information in company if they do not have authorization..

Usability

All contracts and user informations are held on the system. So admins and sales managers can reach any informations easily. The contracts that are made for other customers are held on the system. Not in hard copy.

Performance

Because there is no image file held in the database, fetch operations will made much more faster.The response time should be fast for queries.

Simplicity

The website is very simple to use. The sales manager and admin can be able to see everything easily

3.3System Models

Use Case Model

The main use cases of the program shwown in the use case diagram below.

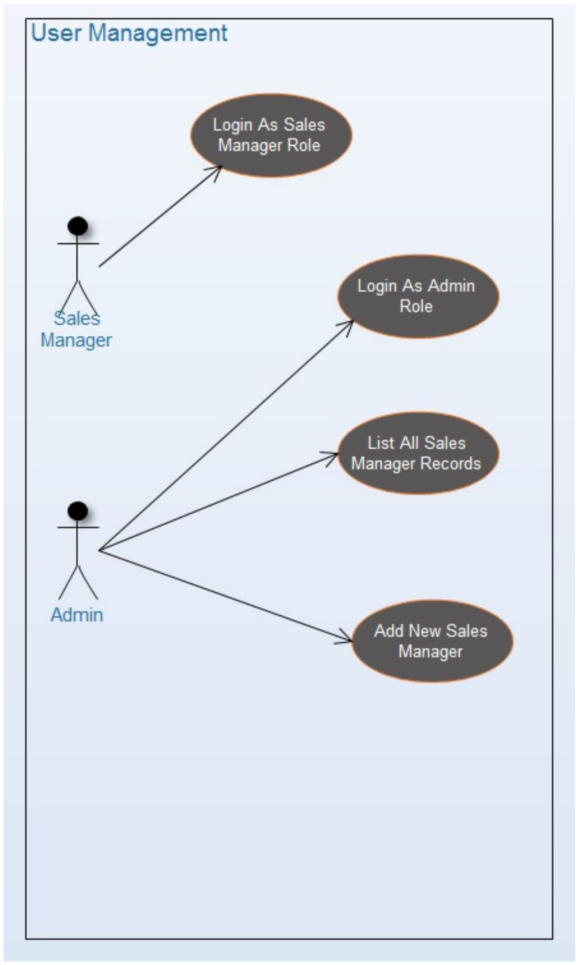

Figure 1 : User Management UseCase Diagram

,

- Sales manager or admin login in system with username and password.
- Admin can add new sales manager.
- Admin can see all sales managers records.
- Admin can delete sales manager with T.C. number

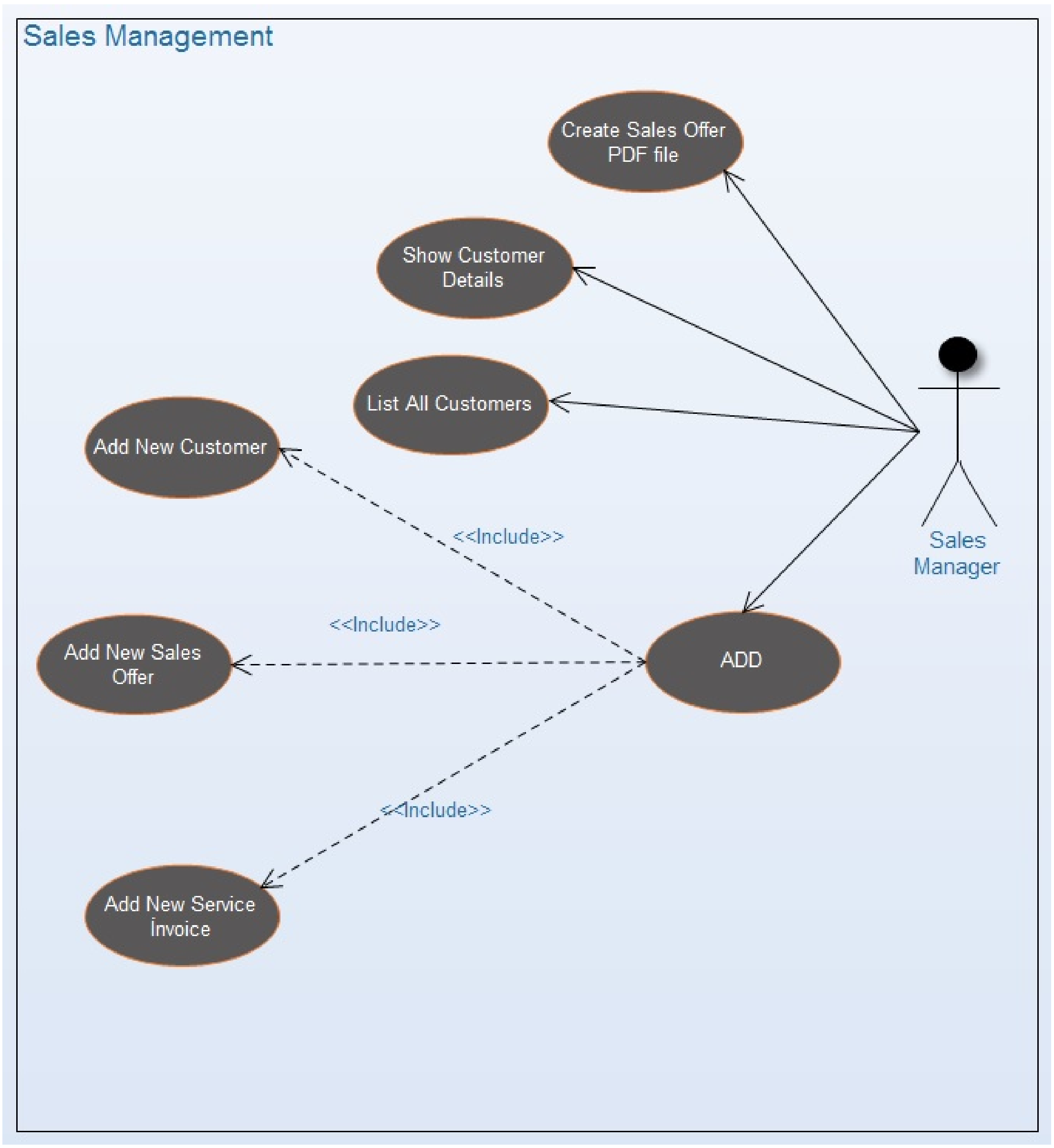

Figure 2 : Sales Management UseCase Diagram

- Sales manager can add with selection criteria.
- Sales manager can see all customers and customers details.
- Sales manager can uptate or delete all customers.
- Sales manager can create sales offer in PDF file

Contex Diagram

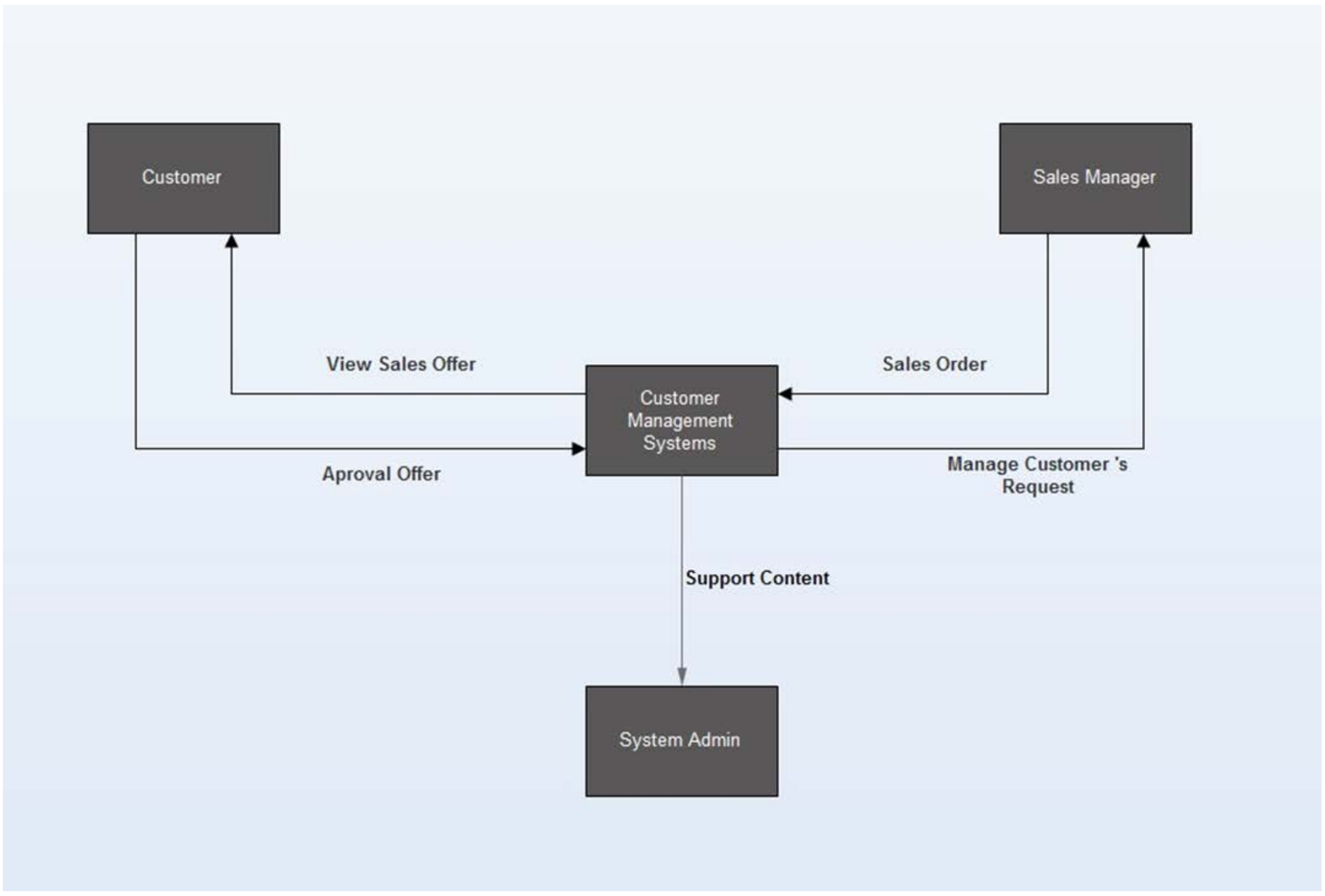

Figure 3 : Customer Tracking System Contex Diagram

4.References

[1] Diagrams, https://www.edrawsoft.com

[2] MVC, https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

[3] SQL, https://en.wikipedia.org/wiki/SQL#See_also

[4] CSS, https://en.wikipedia.org/wiki/Cascading_Style_Sheets

[5] HTML, https://en.wikipedia.org/wiki/HTML