

**CS411 – Software Architecture Design
Final Project
Group 10
Customer Relationship Management System**

**Ali Ozcan
20700687**

**Fuat Basik
20701411**

**M. Yusuf Ertekin
20702750**

**M. Emre Nevayeshirazi
20701946**

Customer Relationship Management System

Table of Contents

Table of Figures	4
1. Introduction.....	6
2. Case Description.....	7
3. Software Architecture Design Process.....	8
4. Requirement Analysis.....	9
4.1. Stakeholders.....	9
4.2. Textual Requirements	10
4.3. Use Cases.....	14
4.4. Textual Use Cases.....	19
4.5. Architectural Scenarios.....	24
4.6. Prototypes	26
5. Technical Problem Analysis	29
5.1. Stakeholder Requirements.....	29
5.2. Computer Science Problems.....	29
5.3. Mathematical Problems	29
5.4. Quality Requirements	30
5.5. Problem Representation for Important Problems.....	30
6. Domain Analysis	35
7. Software Architecture Design	40
8. Top Level Context Diagram	41
9. Module Views	42
9.1. Decomposition View.....	42
9.1.1. Decomposition View Context Diagram.....	42
9.1.2. Decomposition View for Mezigit CRM.....	43
9.2. Uses View.....	44
9.2.1. Uses View Context Diagram.....	44
9.2.2. Uses View for Mezigit CRM.....	45
9.3. Generalization View.....	47
9.4. Layer View and Context Diagram for Layer View.....	48
9.5. Aspect View.....	49
9.6. Data Model View	50
10. Component and Connector Styles.....	52
10.1. Blackboard View.....	52

10.2.	Three Tier View	53
10.3.	Pipes and Filters View	54
10.4.	Cloud View	54
11.	Allocation Views	56
11.1.	Deployment View	56
11.1.1.	Deployment View Context Diagram	56
11.1.2.	Deployment View for Mezgit CRM.....	56
11.2.	Install View.....	57
11.3.	Work Allocation View.....	58
12.	Software Architecture Evaluation.....	59
12.1.	Scenario Based Architecture Analysis Method.....	59
12.1.1.	Description of the Candidate Architecture	59
12.1.2.	Development and Prioritization of Scenarios	60
12.1.3.	Scenario Evolution.....	61
12.1.4.	Reveal Scenario Interactions	62
12.1.5.	Overall Evaluation	62
12.2.	Architecture Trade-off Analysis Method.....	64
13.	Conclusion.....	66
14.	References.....	67

Table of Figures

Figure 1 Use Case Diagram.....	15
Figure 2 Use Case Diagram of Administrator.....	16
Figure 3 Use Case Diagram of Company User.....	17
Figure 4 Use Case Diagram of Customer.....	18
Figure 5 Use Case Diagram of Moderator.....	19
Figure 6 Main Screen Mockup.....	26
Figure 7 Opportunities Screen Mockup.....	27
Figure 8 Reports Screen Mockup.....	28
Figure 9 Software Architecture Design.....	40
Figure 10 Top Level Context Diagram.....	41
Figure 11 Decomposition View Context Diagram.....	42
Figure 12 Decomposition View.....	43
Figure 13 Context Diagram for Uses View.....	44
Figure 14 Uses View.....	45
Figure 15 Generalization View.....	47
Figure 16 Layer View and Context Diagram for Layer View.....	48
Figure 17 Aspect View.....	49
Figure 18 Data Model Style.....	50
Figure 19 Blackboard View.....	52
Figure 20 Three Tier View.....	53
Figure 21 Pipes and Filters View.....	54
Figure 22 Cloud View.....	54
Figure 23 Context Diagram for Deployment View.....	56
Figure 24 Deployment View.....	56
Figure 25 Install View.....	57
Figure 26 Work Allocation View.....	58
Figure 27 Candidate Architecture.....	59

List of Tables

Table 1 Sub-Problem Table	35
Table 2 Overall Knowledge Sources.....	35
Table 3 Overall Knowledge Source Evaluations.....	35
Table 4 Security Knowledge Source.....	36
Table 5 Security Knowledge Sources Evaluation	36
Table 6 User Interface Sources	36
Table 7 User Interface Sources Evaluation	36
Table 8 Data Storage Sources	36
Table 9 Data Storage Sources Evaluation	37
Table 10 Cloud Computing Sources	37
Table 11 Cloud Computing Sources Evaluation	37
Table 12 Business Forecasting Sources	37
Table 13 Business Forecasting Sources Evaluation	37
Table 14 Reliability Sources	38
Table 15 Reliability Sources Evaluation.....	38
Table 16 Consistency Sources	38
Table 17 Consistency Sources Evaluation	38
Table 18 Solution Domain Concepts	39
Table 19 Uses View Matrix Representation	46
Table 20 Developments and Prioritization of Scenarios	60
Table 21 Scenario Evolution	61
Table 22 Reveal Scenario Interactions Part 1.....	62
Table 23 Reveal Scenario Interactions Part 2.....	62
Table 24 Utility Table	65

1. Introduction

Customer Relationship Management (CRM) software solutions allow a company to manage the relationships company have with its' customers, using a combination of people, processes and technology.[2] It helps companies to learn more about their customers and customers behavior so they can develop stronger relationships that last longer with the customers.

Key stability in today's dynamic marketplace is to attain and retain customers. As it is in the past, companies cannot hold in the market by only offering some product or service to their customers. In addition to these, companies must have good relationship with their customers in order to have long-term relationships with them. Besides, while keeping existing customers, getting new customers is an essential issue in today's business world. Therefore, satisfying customer needs become a vital issue for companies.

In order to solve this problem, Customer Relationship Management strategy must be adapted by companies. Customer Relationship Management is a company strategy that is essential for companies to reduce costs and increase efficiency while having good relationships with customers. The CRM approach can be automated as CRM computer software. CRM software should be able to aid companies in terms of marketing, sales, customer support and technical support. To provide these features CRM software brings information from all data sources within an organization and sometimes even from outside the organization [1]. This is essential for companies, since according to these data correct decisions can be taken to meet customer needs. This helps companies to develop stronger relationships with customers.

Remaining parts of document focuses on the case description, software architecture design process, requirement analysis, technical problem analysis, domain analysis, top level context diagram, module views, component and connector views, allocation views and evolution of software architecture.

2. Case Description

Increasing technology has caused an exponential rate growth by 2003 and total amount of commerce conducted over web had top \$3 trillion [3]. Thanks to electronics business, merchants can get to market faster with new products and services. This fastening at the marketing has increased number of customer and also it increased complexity of customer relationships. This complexity not only changed also way of companies to market their products/services but also created new problems for the companies waiting to be solved.

Customer Relationship Management (CRM) systems are mainly designed for the companies, therefore, system is designed to solve companies' problems and provide solutions to their needs.

To have a single problem statement, it will be good to mention most common problems that companies face in this manner. These problems can be listed as:

- Having products or services but having problems to reaching to target market.
- Not having a particular way to market products or service to the prospect customers.
- Unsatisfied customers who not buy service or products from same company again. [3]

Following need of the companies also created the need of the CRM systems:

- Need to simplicity and ease of integration.
- Need to remote access to company data.
- Need to mobile access to company data.
- Need to integrated analytics.
- Need to list of data management.
- Need to flexibility and customization.
- Need to leading generation and following up tracking.

Problems and needs listed above describe the current situation of the market and companies and they show why we need CRM systems.

3. Software Architecture Design Process

Software architecture design process consists of three essential parts. These are Technical Problem Analysis, Solution Domain Analysis and Alternative Space Analysis. To combine these three with the “Mezgit”, it is good to mention these steps one by one.

Technical Problem Analysis: Technical problem analysis is a systematic process that is conducted to define what the essential problems of the system are. After finding the essential needs of the system, it will be possible to define problems, identify sub-problems, specify sub-problems and prioritize them. So we can say that, technical problem analysis will create a link from requirements to the technical problems. In the “Mezgit” to analyze problems technically, analyze of the domain is a necessity. Since CRM systems requires knowledge at marketing and business areas, technical problem analysis of these system will be depend on mostly on the domain analysis. During these analysis steps, our problem description and sub-problems will help to define our domain scope and domain analyze will help to define problem and sub-problem descriptions. There will be iteration between these two steps and at the end of the technical problem analysis technical problems and sub-problems of the project will be defined.

Solution Domain Analysis: Solution domain analysis is the systematic activity of collecting, organizing and storing domain knowledge. It will include a solution concept independent from the requirements. After the technical problem analysis, in other words after reaching technical problem definition and also technical sub-problems, for each of these problems domain scope will be defined. With the help of this scope domain knowledge sources will also defined. In the “Jobs Project” these sources will be mainly domain literature and existing systems. For the each sub-problem solution domains will be defined and prioritized, for each domain knowledge source will identified and prioritized, from each knowledge source domain concepts will be extracted and solution domain concepts will structure and refined. In other words all the technical problems and sub-problems will be mapped to the domain.

Alternate Space Analysis: While doing domain analysis, alternate solutions also will be determined and in the case using one solution creates a conflict with any of other solutions, alternative solution will be used. Not any extra work will be done for this part but during the domain analysis, alternate space analysis also will be taken into account.

4. Requirement Analysis

4.1. Stakeholders

End Users

- **System Administrator:** Responsible for general usage of the system. Shapes the CRM system through its life with no permissions.
- **Moderator:** Is an instance of system administrator. Responsible for general usage of the system but with lower permission.
- **Company User:** Responsible for general usage of the system.
- **CRM Customer:** Responsible for general usage of the system.

Customer: Pays for the system and ensures its delivery.

Developers

- **Database Administrator:** Involved in many aspects of the data stores, including database design, data analysis, data modeling and optimization, installing of the database software, and monitoring and administration of database security.
- **Implementer:** Responsible for the development of specific elements according to designs, requirements, and the architecture.
- **Integrator:** Responsible for taking individual components and integrating them according to the architecture and system designs.
- **System Engineer:** Responsible for design and development of system components in which software plays a role.
- **Tester:** Responsible for the test and verification of the system or its elements against the formal requirements and the architecture.

Architect: Responsible for the development of the architecture and its documentation.

Conformance Checker: Responsible for assuring conformance to standards and processes to provide confidence in CRM Products suitability.

Deployer: Responsible for accepting the completed system from the development effort and deploying it, making it operational, and fulfilling its allocated business function.

Designer: Responsible for systems and design downstream of the architecture, applying the architecture to meet specific requirements of the parts for which they are responsible.

Evaluator: Responsible for conducting a formal evaluation of the architecture against some clearly defined criteria.

Maintainer: Responsible for fixing bugs and providing enhancements to the system throughout its life.

Project Manager: Responsible for planning, sequencing, scheduling and allocating resources to develop software components and deliver components to integration and test activities.

4.2. Textual Requirements

System Administrator

- System should be able to allow administrator to add new users to the system.
- System should be able to allow administrator to remove users from system.
- System should be able to allow administrator to modify existing users.
- System should be able to allow administrator to create new users groups to the system.
- System should be able to allow administrator to assign moderators to user groups.
- System should be able to allow administrator to set permission of groups.
- System should be able to allow administrator to change permission of groups.
- System should be able to allow administrator to remove moderators from user group.
- System should be able to allow administrator to add new products to the system.
- System should be able to allow administrator to remove products from system.
- System should be able to allow administrator to modify product information.
- System should be able to allow administrator to create goals for company.
- System should be able to allow administrator to modify existing goals.
- System should be able to allow administrator to delete existing goals.
- System should be able to allow administrator to assign goals to moderators.

Moderator

- System should be able to allow moderators to add users to their own group.
- System should be able to allow moderators to remove users from their own group.

- System should be able to allow moderators to modify users' from their own group.
- System should be able to allow moderators to assign goals to users under their control.

Company User

- System should be able to allow company users to should be able to create new contacts.
- System should be able to allow company users to modify contacts created by them.
- System should be able to allow company users to delete contacts created by them.
- System should be able to allow company users to view data about contacts created by them or other users.
- System should be able to allow company users to create opportunities from their contacts.
- System should be able to allow company users to send individual email to their or other user's contacts.
- System should be able to allow company users to send mass email to their or other user's contacts.
- System should be able to allow company users to create accounts on system for their contacts.
- System should be able to allow company users to delete accounts of contacts that are added system by them.
- System should be able to allow company users to export contacts as an Excel file.
- System should be able to allow company users to create new leads.
- System should be able to allow company users to modify leads created by them.
- System should be able to allow company users to delete leads created by them.
- System should be able to allow company users to view data about leads created by them or other users.

- System should be able to allow company users to convert their leads to contacts.
- System should be able to allow company users to create opportunities from their leads.
- System should be able to allow company users to send individual email to their or other user's leads.
- System should be able to allow company users to send mass email to their or other user's leads.
- System should be able to allow company users to export leads as an Excel file.
- System should be able to allow company users to create appointments on Calendar.
- System should be able to allow company users to delete appointments on Calendar.
- System should be able to allow company users to modify appointments on Calendar.
- System should be able to allow company users to comment on their appointments or other company user's appointments.
- System should be able to allow company users to add activities to the system.
- System should be able to allow company users to remove activities from the system.
- System should be able to allow company users to modify their activities.
- System should be able to allow company users to view activities that are created by them or by other company users.
- System should be able to allow company users to comment on activities that are created by them or by other company users.
- System should be able to allow company users to create folders.
- System should be able to allow company users to delete folders.
- System should be able to allow company users to modify folders information.
- System should be able to allow company users to add documents to system.
- System should be able to allow company users to remove their documents.
- System should be able to allow company users to assign their documents to folders.

- System should be able to allow company users to associate documents with their contacts if contact has system account.
- System should be able to allow company users to share their documents with other users.
- System should be able to allow company users to view and download documents that are created by them or other users.
- System should be able to allow company users to create opportunities.
- System should be able to allow company users to remove opportunities created by them.
- System should be able to allow company users to modify opportunities created by them.
- System should be able to allow company users to export opportunities as Excel File.

Contact Accounts

- System should be able to allow contacts to view and download the documents associated with them.
- System should be able to allow contacts to comment on documents associated with them.
- System should be able to allow contacts to delete their comments from documents associated with them.
- System should be able to allow contacts to add comments about product.
- System should be able to allow contacts to modify comments about product.
- System should be able to allow contacts to delete their comments about product.
- System should be able to allow contacts to enter their request to the system by filling request form.
- System should be able to allow contacts to enter their complaints to the system about product by filling complaint form.

4.3. Use Cases

In a CRM system, system administrator has the highest authorization among all stakeholders. Moderator is an instance of system administrator; it has a bit less authorization. Company user is an ordinary user which has rarely no permissions among them. Among these users, user capabilities go up from company user to system administrator. Therefore, instead of drawing use-case diagrams separately from user to user. It is shown that in a single figure. In the following page Use Case Diagram can be found.

Here are the separate use case diagrams for each user.

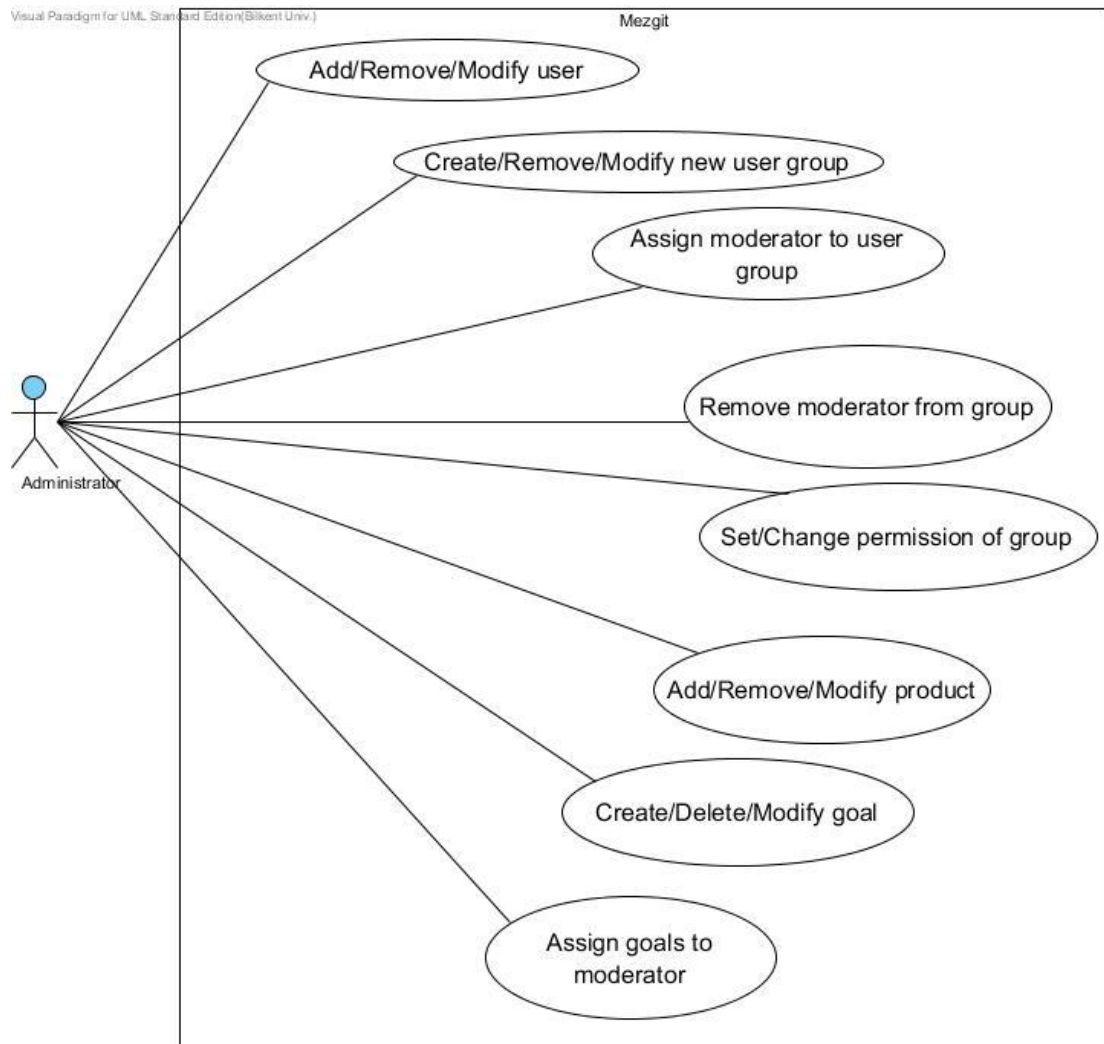


Figure 2 Use Case Diagram of Administrator

Administrators have the cardinal rights on Mezgit system. They are responsible base operations about all Mezgit's users. They manage existence of users and determine user groups and their permissions according to user information. They determine goals and assign them to moderators. Administrators have important roles on Mezgit because they have permission to do all operations in system.

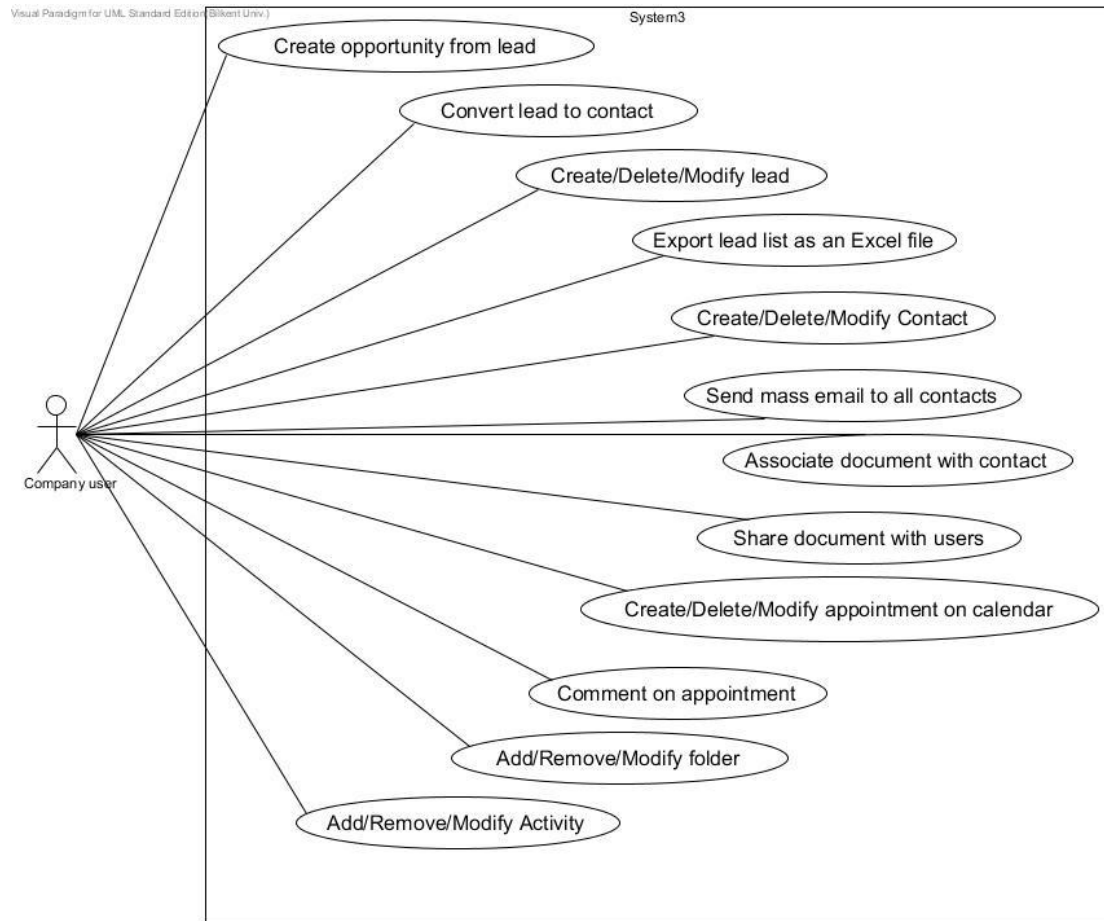


Figure 3 Use Case Diagram of Company User

Company users are mostly interested with contacts and their operations. In addition, they are responsible to analyze leads and get possible opportunities from leads. They manage activities on the system and manage contacts with workers by managing appointment and documents.

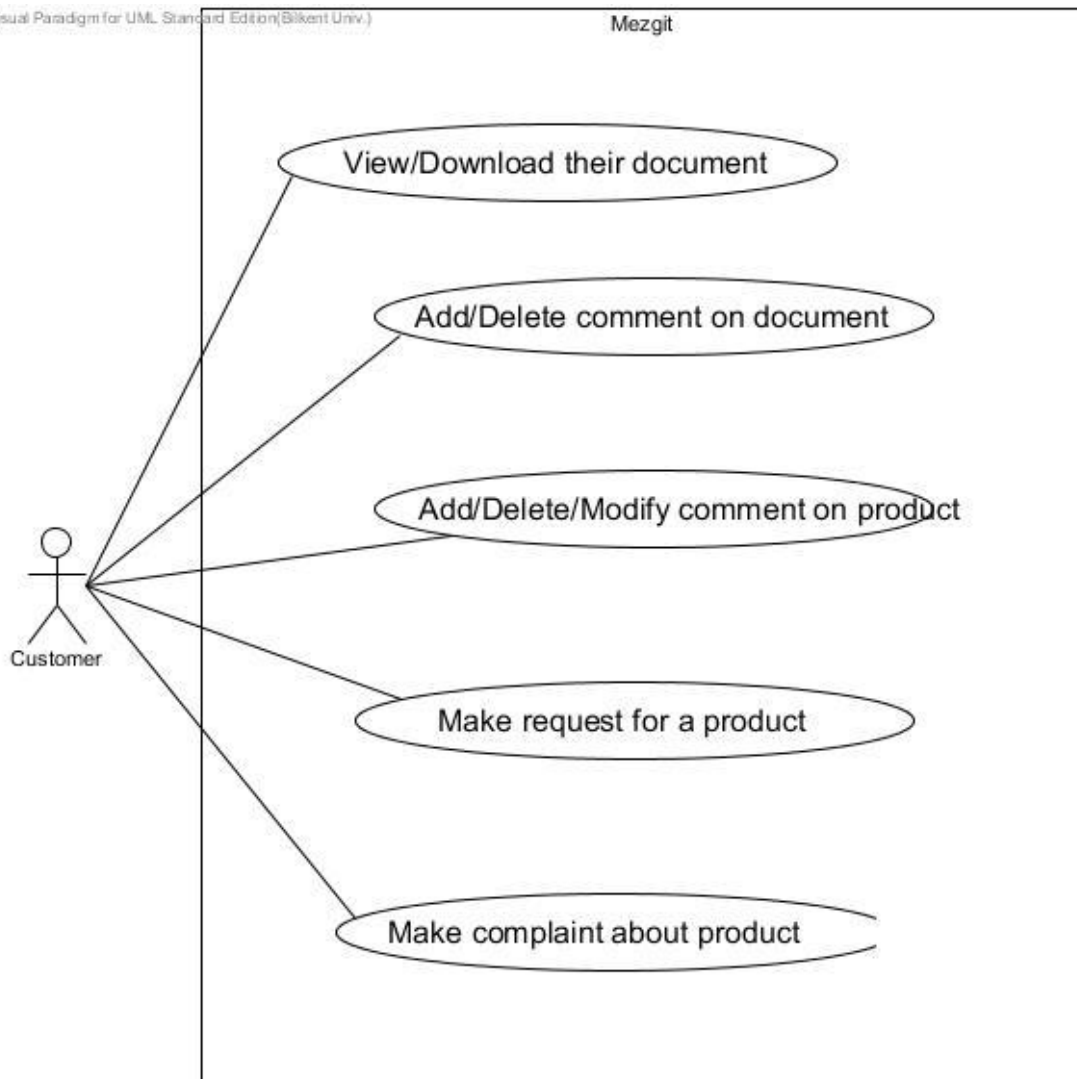


Figure 4 Use Case Diagram of Customer

Customers are highly interested with products information and their qualities. They have right to make complain about a product to inform company to fix it. In addition, they can request for a new product so that company can get feedback from the field. They check out related documents and make comment on them.

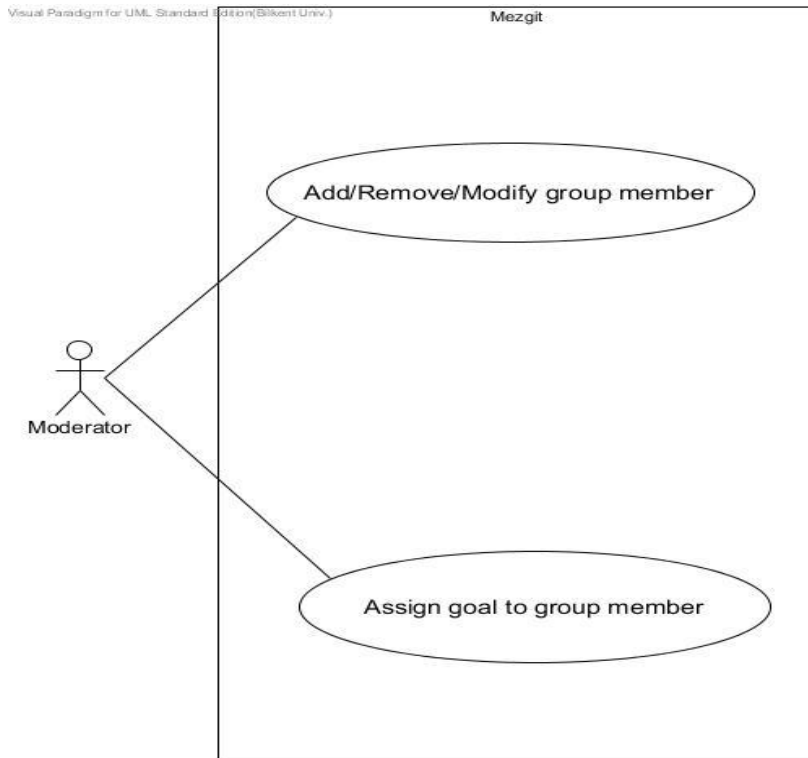


Figure 5 Use Case Diagram of Moderator

Moderators are highly responsible on their groups that are assigned them by administrator. They can manage all operations related with their group. Although, moderator seems likely be to do only two operations, they can do also company user, and customer' operations because of prioritization feature of Mezgit.

4.4. Textual Use Cases

Use Case Name: Assign moderator to user group.

Participating Actors: Administrator: Actor who has most permission to use system.

Entry Condition: Administrator right clicks on a user group and selects “Assign moderator”.

Flow of Events:

- 1-The end user right clicks on a user group.
- 2-Options related to user group shows up.
- 3-He selects “Assign moderator”.
- 4-List of existing moderators will appear.
- 5-He selects moderator that he wants to assign for user group.

Exit Condition: The use case is terminated when administrator clicks “Apply”.

Special Requirements: At least one moderator should exist in system.

Use Case Name: Assign goals to moderator.

Participating Actors: Administrator: Actor who has most permission to use system.

Entry Condition: Administrator right clicks on one moderator.

Flow of Events:

1-Administrator right clicks on one moderator.

2-Options related to moderator show up.

3-He selects “Assign goal”.

4-List of existing goals will appear.

5-He selects goal that he wants to assign for moderator.

Exit Condition: This use case ends when administrator clicks “Apply”.

Special Requirements: There should be at least one goal in system.

Use Case Name: Assign goals to group member.

Participating Actors: Administrator: Actor who has most permission to use system.

Moderator: Actor who is responsible for a specific group.

Entry Condition: User right clicks on one group member.

Flow of Events:

1-User right clicks on one group member.

2-Options related to group member show up.

3-He selects “Assign goal”.

4-List of existing goals will appear.

5-He selects goal that he wants to assign for group member.

Exit Condition: This use case ends when user clicks “Apply”.

Special Requirements: There should be at least one goal in system.

Use Case Name: Convert lead to contact.

Participating Actors: Administrator: Actor who has most permission to use system.

Moderator: Actor who is responsible for a specific group.

Company user: Actor who is an ordinary company employee.

Entry Condition: User opens list of leads (possible contacts).

Flow of Events:

- 1-User opens list of all leads.
- 2-He right clicks on lead that he wants to convert to contact.
- 3-Options related to lead show up.
- 4-He selects “Convert it to contact”.
- 5-Selected lead automatically moves to contact list.

Exit Condition: This use case ends when user clicks “Convert it to contact”.

Special Requirements: There should be at least one lead in system.

Use Case Name: Create opportunity from lead.

Participating Actors: Administrator: Actor who has most permission to use system.

Moderator: Actor who is responsible for a specific group.

Company user: Actor who is an ordinary company employee.

Entry Condition: User opens list of leads.

Flow of Events:

- 1-User opens list of all leads.
- 2-He right clicks on lead that he wants to create opportunity related to it.
- 3-Options related to lead show up.
- 4-He selects “Create opportunity”.
- 5-A form shows up.
- 6-He fills up the form with required information about opportunity.
- 7-New opportunity related to lead is added to opportunity list.

Exit Condition: This use case ends when user clicks “Apply” on form.

Special Requirements: There should be at least one lead in system.

Required fields of the form must be filled.

Use Case Name: Export contact list as an Excel file.

Participating Actors: Administrator: Actor who has most permission to use system.

Moderator: Actor who is responsible for a specific group.

Company user: Actor who is an ordinary company employee.

Entry Condition: User right clicks on empty space of contact list.

Flow of Events:

- 1-User opens list of contacts.
- 2-He right clicks on empty space of contact list.
- 3-Options related to contact list show up.
- 4-He selects "Export contact list".
- 5-He selects "Excel" option.
- 6-Directory selection screen shows up.
- 7-He enters directory that he wants to save excel file.

Exit Condition: This use case ends when user clicks "Save".

Special Requirements: There should be free space on directory that user wants to save on.

There should be at least one contact.

Use Case Name: Associate document with contact.

Participating Actors: Administrator: Actor who has most permission to use system.

Moderator: Actor who is responsible for a specific group.

Company user: Actor who is an ordinary company employee.

Entry Condition: User creates new document.

User right clicks on existing document.

Flow of Events:

- 1-User selects "Add new document"
- 2-A form shows up.
- 3-User fill require fields of form.
- 4-If he prefers to associate document with contact at the creation time, he clicks "Associate with contact" option.
- 5-Contact list shows up.
- 6-He selects contact that he wants to associate document with.
- 7-User right clicks on existing document.
- 8-Options related to document show up.
- 9-He selects "Associate with contact".
- 10-Contact list shows up.
- 11-He selects contact that he wants to associate document with.

Exit Condition: This use case ends when user clicks "Apply".

Special Requirements: There should be at least one contact in system.

Use Case Name: Send mass email to contacts.

Participating Actors: Administrator: Actor who has most permission to use system.

Moderator: Actor who is responsible for a specific group.

Company user: Actor who is an ordinary company employee.

Entry Condition: User clicks send email.

Flow of Events:

1-User clicks send email.

2-Email screen shows up.

3-He selects all contacts for receiver.

4-He writes subject.

5-He writes mail.

Exit Condition: This use case ends when user clicks “Send”.

Special Requirements: There should be at least one contact in system.

Use Case Name: Make request for a product.

Participating Actors: Administrator: Actor who has most permission to use system.

Moderator: Actor who is responsible for a specific group.

Company user: Actor who is an ordinary company employee.

Customer: Actor who has account on system as a contact.

Entry Condition: User clicks on “Make request”.

Flow of Events:

1-User clicks on “Make request”.

2-Request form shows up.

3-He writes subject of the request.

4-He writes name of the product that he request.

5-He writes description of product.

Exit Condition: This use case ends when user clicks “Send”.

Use Case Name: Make complaint about product.

Participating Actors: Administrator: Actor who has most permission to use system.

Moderator: Actor who is responsible for a specific group.

Company user: Actor who is an ordinary company employee.

Customer: Actor who has account on system as a contact.

Entry Condition: User clicks on “Make complaint”.

Flow of Events:

1-User clicks on “Make complaint”.

2-Complaint form shows up.

3-He writes name of the product that he wants to complaint about it.

4-He writes reasons of the complaint.

Exit Condition: This use case ends when user clicks “Send”.

4.5. Architectural Scenarios

Adding a new employee to CRM

Berk Arslan is recent employee of a company. His CRM account is going to be open. System administrator Abidin Berkay Simsek will open a CRM account for Berk. Abidin opens a web browser, then types the Mezgit CRM web service. He login to the system with user-id and password. He creates a new account. Then, administrator fills the required fields including name, surname, date of birth, date of graduation, level of education and password according to Berk’s information. Then he finishes to adding Berk to the system.

Adding new employee to a group:

The company has a new employee named Berk Arslan. He is assigned to a software group. The group has a CRM group and Berk needs a membership to put his works to group. System administrator System administrator Okan Cetin will open a CRM account for Berk. Okan opens a web browser, and then types the Mezgit CRM web service. He login to the system with user-id and password. Okan Cetin goes to group tab and selects the group that Berk will be involved. After that, Berk Arslan gets his membership in that group.

New goal is assigned to Berk Arslan:

Berk Arslan's group leader Yagiz Salor assigns him a new goal. Technical Requirement Analysis report needs to be done and group leader need to assign it. Yagiz opens a web browser, and then types the Mezgit CRM web service. He login to the system with user-id and password. He adds a new goal by using appropriate tabs at user interface. After that, he fills the goal description, deadline and selects Berk from employee list.

4.6. Prototypes

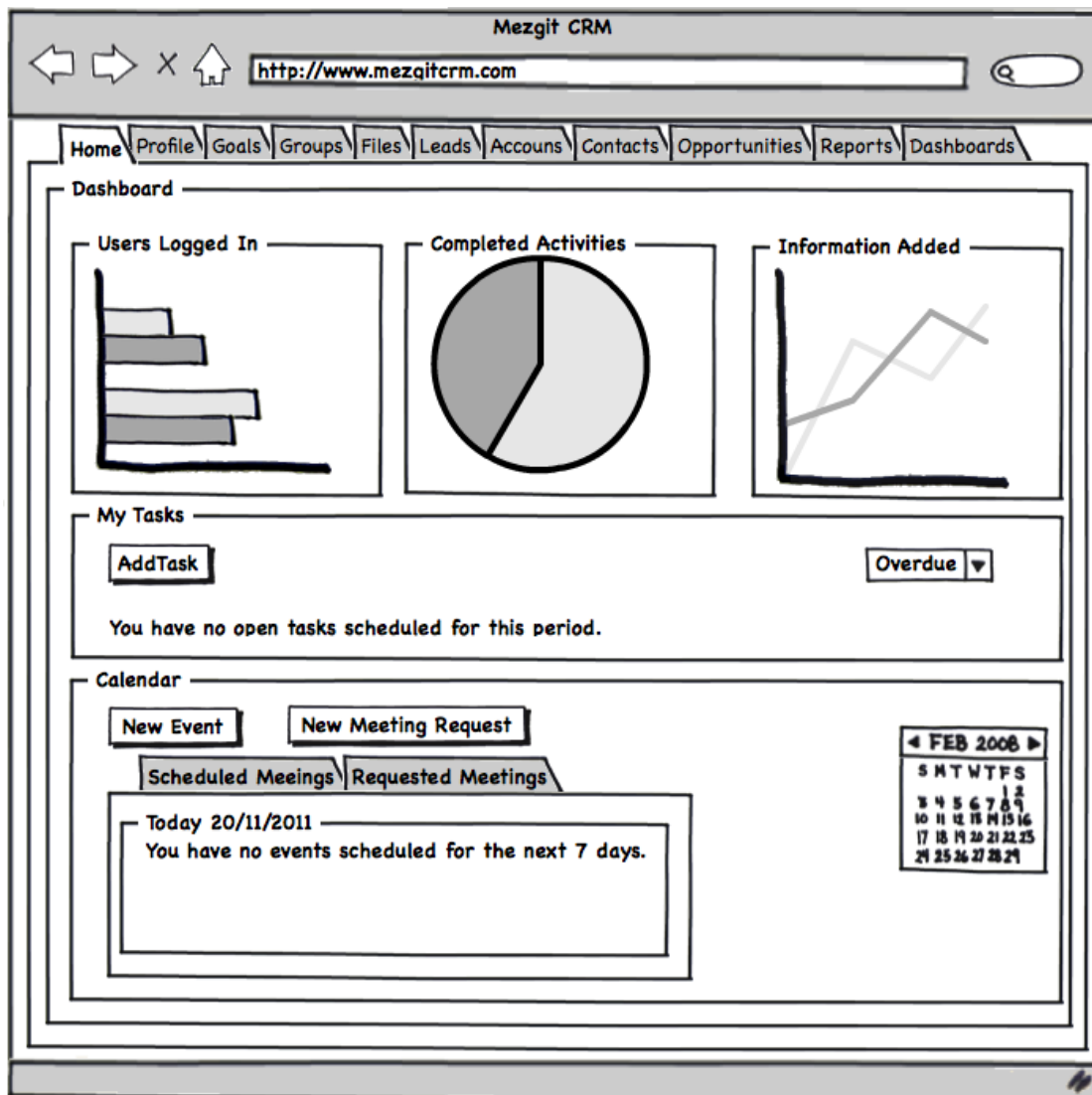


Figure 6 Main Screen Mockup

Figure 6 Main Screen Mockup is the main screen of Mezgit CRM system. There is a dashboard which shows the recent graphs about the company including users logged in, completed activities and information added to the company. In below of these graphs, Task segment appear. In task segment, there exists no overdue task right now. Existing tasks can be seen by selecting type of showing them in combo box. In addition, new task can be assigned to the company just by hitting the add task button. Below task segment, calendar appears. In calendar segment, user can see events and meetings. By selecting scheduled meetings and requested meetings, user can do appropriate work that is needed. Hitting the new event button, user can add new

events or requested meetings to the system. Of course, selecting new meeting date from calendar picker is necessary.

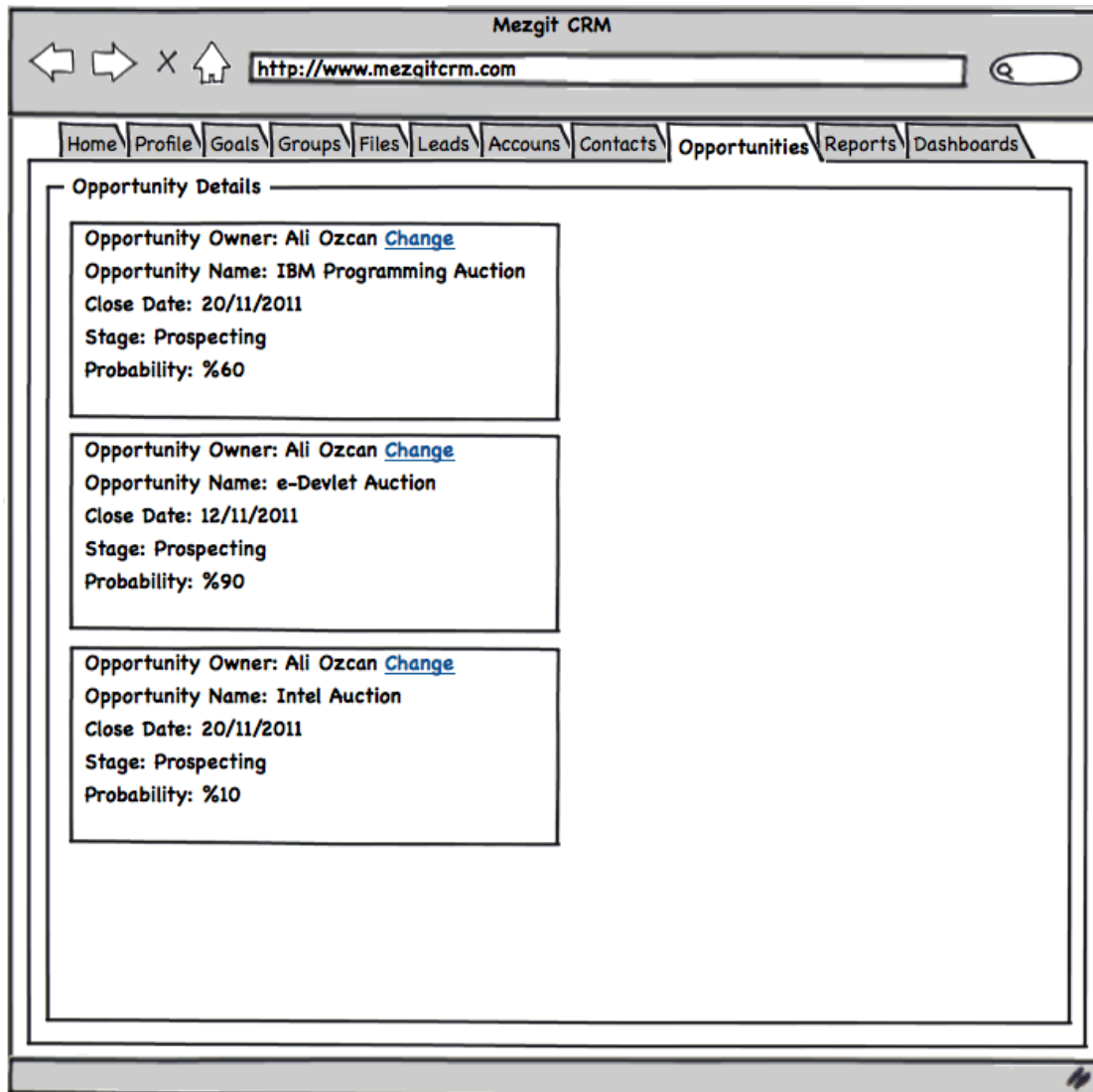


Figure 7 Opportunities Screen Mockup

Figure 7 Opportunities Screen Mockup company users can see existing opportunities. In this figure, opportunities are only auctions which are open some government based companies and Intel and IBM corporations. For each opportunity, there exist opportunity owner, opportunity name, and opportunity closing date, stage of opportunities and lastly, probability of getting a specific auction. Appropriate user can add new opportunities to the system.

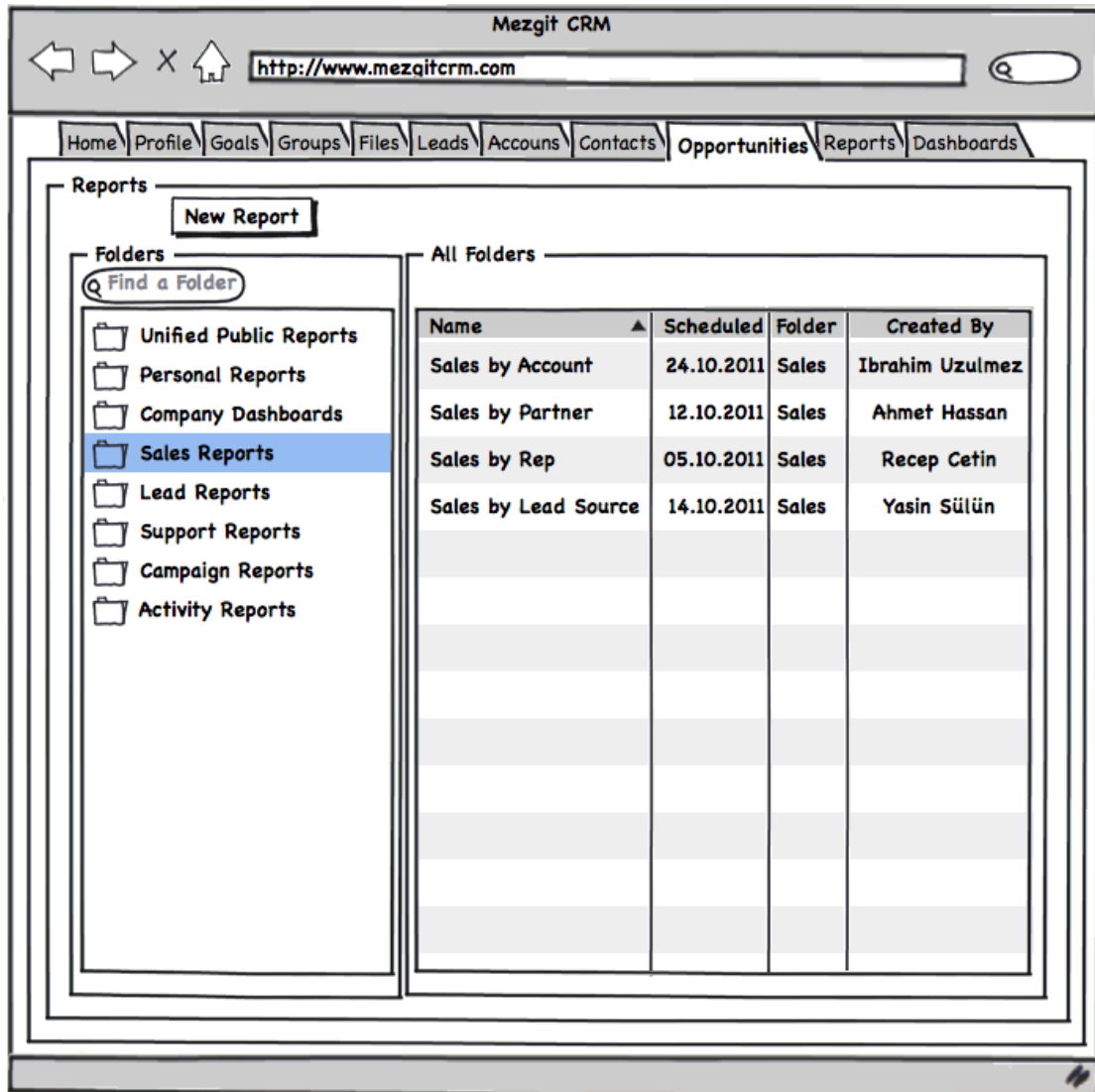


Figure 8 Reports Screen Mockup

Figure 8 Reports Screen Mockup shows the report tab of the “Mezgıt” CRM system. In this tab, reports that are added by users can be seen. On the left side, folders group reports exist. On the right hand side, for a specific folder, written reports can be seen. User can sort these reports according to its name and date.

5. Technical Problem Analysis

5.1. Stakeholder Requirements

Problem 1: How to do manage different user types?

Problem 1.1: How to do authorization: Users of the system should be able to login and authorize into system. That is an essential technical problem that needs to be solved for our system.

Problem 1.2: How to define access rights with respect to user types?

5.2. Computer Science Problems

Problem 1: How to Store Data?

Problem 1.1: How to store textual data: Most of the data that is going to be stored in our application is a textual data such as contacts, their properties and so on. This data is very crucial for our system and effective way to store such data should be investigated.

Problem 1.2: How to store information persistently:

Problem 1.3: How to store files: In addition to textual data, files need to be kept in our system. Effective and easy way for to do so.

Problem 2: How to design user interface?

Problem 2.1: How to define ergonomic user interface: The CRM System that we are designing has lots of important features and properties. Therefore, to be easy to use and provide effectiveness to program users, the user interface must be ergonomic enough.

Problem 2.2: How to define different user interfaces for different user types?

Problem 2.3: How to display data asynchronously?

Problem 3: How to provide security?

Problem 3.1: How to protect user data from being stolen?

Problem 3.2: How to provide security of communication and private data?

5.3. Mathematical Problems

Problem 1: How the CRM system optimization should be?

Problem 2: What is the expected cost for the system?

Problem 3: How to analyze data for forecasts?

5.4. Quality Requirements

Problem 1: How to achieve consistency?

Problem 2: How to provide reliability?

Problem 3: How to provide security?

Problem 4: How to provide safety?

Problem 5: How to provide reliability?

Problem 6: What should be the robustness of the system?

Problem 7: What are the performance requirements?

Problem 8: What are the concepts used to increase reusability?

Problem 9: What are the priorities?

5.5. Problem Representation for Important Problems

Name: How to design architecture for an Internet based application, which can store and exchange information between end users in network environment?

Description: Mezgit CRM software will be a web based database application. Unlike desktop applications there are many different and important concerns in web application. Web applications provide better way to exchange information between to end-users of the system. Hence, good architecture is needed for Mezgit System to cope with problems.

Initial state: The Mezgit can best be compared with CRM sites of today like Microsoft dynamic CRM online. The storage capacity is not as large as it will be for the Mezgit the same can be said for the connectivity aspects, initial systems are only compatible with computers.

Desired state: The Mezgit will be connectable in-home, at-office, on-the-move, that can be used any environment like mobiles, tablets which user can be online. Therefore, Mezgit will make use of cloud computing services.

Urgency/Priority: Very crucial. (1)

Type of Problem: Stakeholder Requirement

Solution Domain: Cloud Computing

Name: How to store content?

Description: CRM systems are all about storing data and using this data effectively to provide better way to cope with today's business problems. Therefore, this technical problem needs to be analyzed and solved in Mezgit CRM System.

Initial state: Most of the today's CRM systems use classical client/server architecture. However, deployment, maintaining and administration of this classical approach are important problems in today's IT world.

Desired state: The Mezgit will use cloud database services to store its content. This will enable developers to focus on application itself instead of building, administering and maintaining databases. Through cloud database Mezgit will be easily adjustable in terms of storage.

Urgency/Priority: Very crucial. (1)

Type of Problem: Computer Science Problem

Solution Domain: Storage

Name: How to display data asynchronously?

Description: HTTP protocol is designed in a request/response manner. Therefore, if data is not transferred between client and server asynchronously, the whole web page need to be refreshed and updated. This causes traffic in network and which is a problem for huge systems. Besides, for use friendly interface, refreshing whole page is not a good idea. Hence, asynchronous transfer needs to be used as much as possible.

Initial state: Although most of the CRM solutions try to display data like regular desktop application, they need to be improved.

Desired state: The Mezgit will use AJAX technology as much as possible to provide asynchronous communication with database. This will make Mezgit user- friendly and easy to use. Therefore, using AJAX will reduce the unrequired information exchange between end-user and server.

Urgency/Priority: Important but system can run without it. (2)

Type of Problem: Computer Science Problem

Solution Domain: Data Transfer

Name: How to design user interface?

Description: User interface is important for any web application. Good user interface is required for end-users to use system effectively.

Initial state: Today's CRM systems try to make use of newest technologies such as CSS 3.0 and HTML 5 to provide better user interface to users. However, there are some problems that need to be solved. For instance, Microsoft Dynamics CRM runs on Internet Explorer only.

Desired state: To cope with user interface problems, Mezgit will be designed and optimized for common web browsers such as Safari, Firefox, Chrome and Internet Explorer. Mezgit will make use of newest technologies such as CSS 3.0 and HTML 5.

Urgency/Priority: System can run live without it. (2)

Type of Problem: Computer Science Problem

Solution Domain: User Interface

Name: How to do manage different user types?

Description: System administrator, moderator and company user are the different user types which can interact with the Mezgit CRM. These user types have all different authorization levels and capabilities. Mezgit CRM system should handle this change appropriately.

Initial State: Current CRM systems have one different type of user which is only end user. However, it is difficult to use CRMs with single user type.

Desired State: Mezgit CRM system will have four different types of users which are system administrator, moderator, customer and company user. These all are have different privileges.

Urgency/Priority: Priority of this problem is crucial. Since four user types can access to the system. Managing user types must be done in early phases. (1)

Type of Problem: Stakeholder, Computer Science

Sub-Problems: How to do authorization
 How to define access rights with respect to user types?

Solution Domain: Authorization

Name: How to provide security?

Description: In order to serve system with no problem, security issues should be solved. In addition to this, there exist some features like instant messaging and

emailing; therefore, user privacy should be protected. Therefore, Mezgit CRM systems reliability will be very high.

Initial State: It is possible to break security systems of CRM machines because the system is very complex and a crank might occur at sometime. For instance, web browsers allow people to write and run JavaScript code to address bar. It might lead some security problems.

Desired State: Mezgit CRM system provides existing security protocols. In addition to this, Mezgit CRM will prevent some web browser based injections.

Urgency/Priority: Priority of this problem is crucial. (1)

Type of Problem: Computer Science Problem, Mathematical Problem

Sub-Problems: How to protect user data from being stolen?

How to provide security of communication and private data?

Solution Domain: Security

Name: How to define different user interfaces for different user types?

Description: Mezgit CRM system has four different type of users. Each user has different privileges. Therefore, appropriate interface should be shown for each user.

Initial state: While Mezgit CRM system provides different user interfaces for different users, it should be effective and ergonomic. However, cost of this design should not be high.

Desired state: Mezgit CRM system will make use of role based authorization management. This role based authorization will also be used for determining what to display.

Urgency/Priority: Important but not urgent (2).

Type of Problem: Stakeholder Requirement, Computer Science Problem

Solution Domain: User Interface

Name: How to forecast content?

Description: Business forecasting is very important in today's marketing strategies. The reason is that patterns such as seasonality and trend play a crucial role for companies. Besides, analyzing existing data is helpful for sales and marketing, which are the main concerns of CRM.

Initial State: It is possible to compose a CRM system which stores only the detailed work information of the company. The system can also provide some graphical information about work done.

Desired State: CRM system should provide necessary information to company and lead company to shape its future works. By looking at the works the company done and by analyzing them, CRM system should offer the optimal solution for a future work.

Urgency/Priority: Priority of this problem is moderate. (2)

Type of Problem: Computer Science Problem, Mathematical Problem

Solution Domain: Business Forecasting

6. Domain Analysis

Sub – Problem	Solution Domain	Priority
How to do authorization ?	Security	1
How to define access rights ?	Authorization	2
How to define ergonomic user interface ?	User Interface	2
How to define different users interfaces for different user types ?	User Interface	2
How to display data asynchronously ?	User Interface	3
How to protect user data from being stolen ?	Security	1
How to provide security of communication and private data ?	Security	1
How to store textual data ?	Data Storage	1
How to store information persistantly ?	Data Storage	1
How to store files ?	Data Storage	1
How to analyze data for forecasts ?	Business Forecasting	2
How to provide reliability ?	Reliability	1
How to achieve consistency ?	Consistency	2

Table 1 Sub-Problem Table

Overall Knowledge Sources

ID	Knowledge Source	Form
KS1	Microsoft Dynamics CRM 4.0	Existing System
KS2	Salesforce CRM	Existing System
KS3	The CRM Handbook : A Business Guide to CRM	Book
KS4	CRM Fundamentals	Book

Table 2 Overall Knowledge Sources

Overall Knowledge Sources Evaluation

Source	Objectivity	Relevance	Abstraction Quality
KS1	Medium	High	Medium
KS2	Medium	High	Medium
KS3	High	High	High
KS4	High	High	High

Table 3 Overall Knowledge Source Evaluations

Security Knowledge Sources

ID	Knowledge Source	Form
KS1	Web Application Security, A Beginner's Guide	Book
KS2	Taking Steps to Ensure CRM Data Security	Article
KS3	Secure Communications: Applications and Management	Book

Table 4 Security Knowledge Source

Security Knowledge Sources Evaluation

Source	Objectivity	Relevance	Abstraction Quality
KS1	High	Medium	Medium
KS2	High	High	High
KS3	High	Medium	Medium

Table 5 Security Knowledge Sources Evaluation

User Interface Sources

ID	Knowledge Source	Form
KS1	Tolga Çapın	Domain Expert
KS2	Designing Interfaces	Book
KS3	The Essential Guide to User Interface Design	Book

Table 6 User Interface Sources

User Interface Sources Evaluation

Source	Objectivity	Relevance	Abstraction Quality
KS1	High	High	High
KS2	Medium	High	Medium
KS3	High	Medium	Medium

Table 7 User Interface Sources Evaluation

Data Storage Sources

ID	Knowledge Source	Form
KS1	Fazlı Can	Domain Expert
KS2	Microsoft SQL Azure Enterprise Application Development	Book
KS3	Information Storage and Management	Book

Table 8 Data Storage Sources

Data Storage Sources Evaluation

Source	Objectivity	Relevance	Abstraction Quality
KS1	High	High	High
KS2	High	High	High
KS3	High	Medium	Medium

Table 9 Data Storage Sources Evaluation

Cloud Computing Sources

ID	Knowledge Source	Form
KS1	Cloud Computing Architectures	Book
KS2	The Computing Storage HandBook	Book

Table 10 Cloud Computing Sources

Cloud Computing Sources Evaluation

Source	Objectivity	Relevance	Abstraction Quality
KS1	High	High	High
KS2	High	High	High

Table 11 Cloud Computing Sources Evaluation

Business Forecasting Sources

ID	Knowledge Source	Form
KS1	Cemal Deniz Yenigün	Doman Expert
KS2	Business Forecasting	Book
KS3	Successful Business Forecasting	Article

Table 12 Business Forecasting Sources

Business Forecasting Sources Evaluation

Source	Objectivity	Relevance	Abstraction Quality
KS1	High	High	High
KS2	High	High	High
KS3	Medium	High	High

Table 13 Business Forecasting Sources Evaluation

Reliability Sources

ID	Knowledge Source	Form
KS1	Software Reliability Methods	Book
KS2	Optimal Structure of Fault Tolerant Software Systems	Article
KS3	System Software Reliability	Book

Table 14 Reliability Sources

Reliability Sources Evaluation

Source	Objectivity	Relevance	Abstraction Quality
KS1	High	High	High
KS2	High	High	High
KS3	High	Medium	Medium

Table 15 Reliability Sources Evaluation

Consistency Sources

ID	Knowledge Source	Form
KS1	Consistency in Software System Development	Article
KS2	Consistency is Key in Web Self-Service	Article

Table 16 Consistency Sources

Consistency Sources Evaluation

Source	Objectivity	Relevance	Abstraction Quality
KS1	High	High	High
KS2	High	High	High

Table 17 Consistency Sources Evaluation

Solution Domain Concepts

Solution Domain	Solution Domain Concepts
Security	Encryption Manager
Authorization	Authorization Manager
User Interface	Web UI Services, Mobile UI Services
Data Storage	Content Manager
Data Process	Transaction Manager
Cloud Computing	Platform as a Service, Database as a Service, Software as a Service
Business Forecasting	Forecast Manager
Reliability	Backup System
Consistency	Backup System

Table 18 Solution Domain Concepts

7. Software Architecture Design

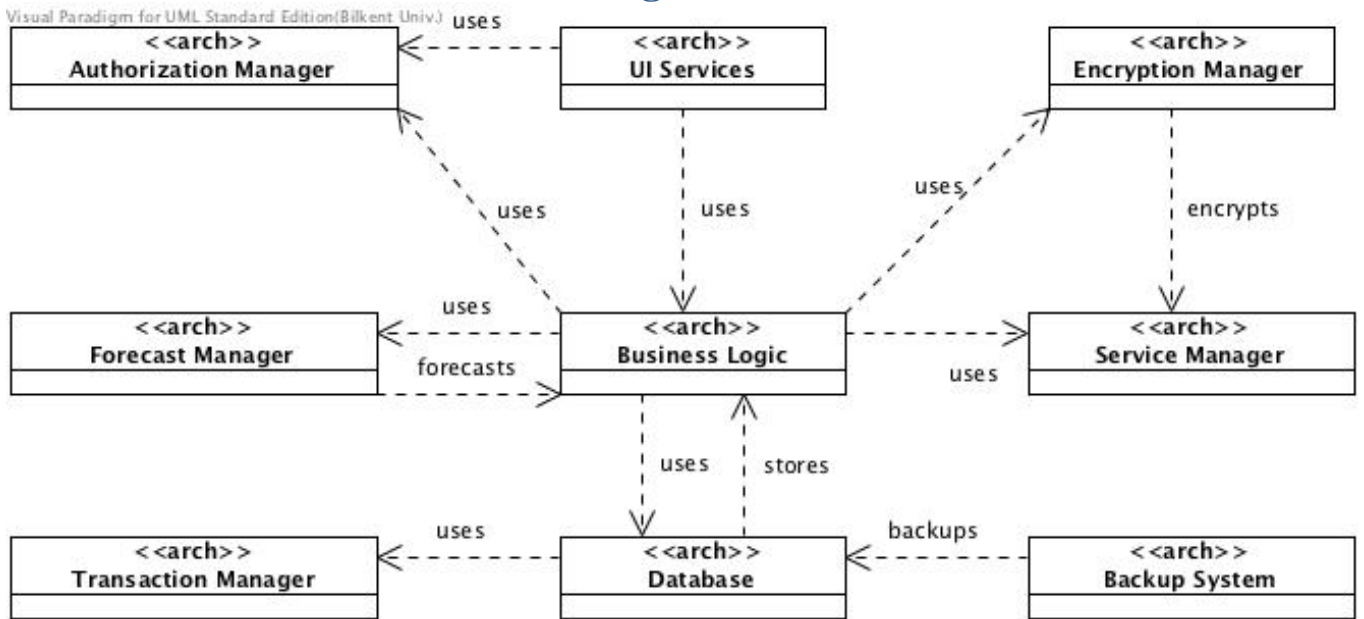


Figure 9 Software Architecture Design

The software architecture design can be also thought as a subsystem decomposition diagram. All the packages are stands for subsystem that have to be implemented.

Among these subsystems, “Backup System” is designed to provide reliability and consistency.

To understand Service Manager, we have to take it into account with database. Since “Mezgit” is a web-based project for the data processing cloud computing necessities should be included. Service Manager and Database are the concepts to provide cloud computing.

Business Logic is the main part of the architecture. It can be thought as brain of the all system that makes calculations, provides communication and also manages the all other modules.

Forecast Manager, is the part of the architecture to provide business forecasting with the help of the Business Logic.

Authorization Manager is the part of the architecture that helps to define authorization for the different user types. UI Manager uses it to decide what type of user entered to the system and it will change the user interface accordingly this type.

Encryption Manager is the security guard of the system. Since, mainly the companies will use the project; security and privacy are the most important topics of the system.

8. Top Level Context Diagram

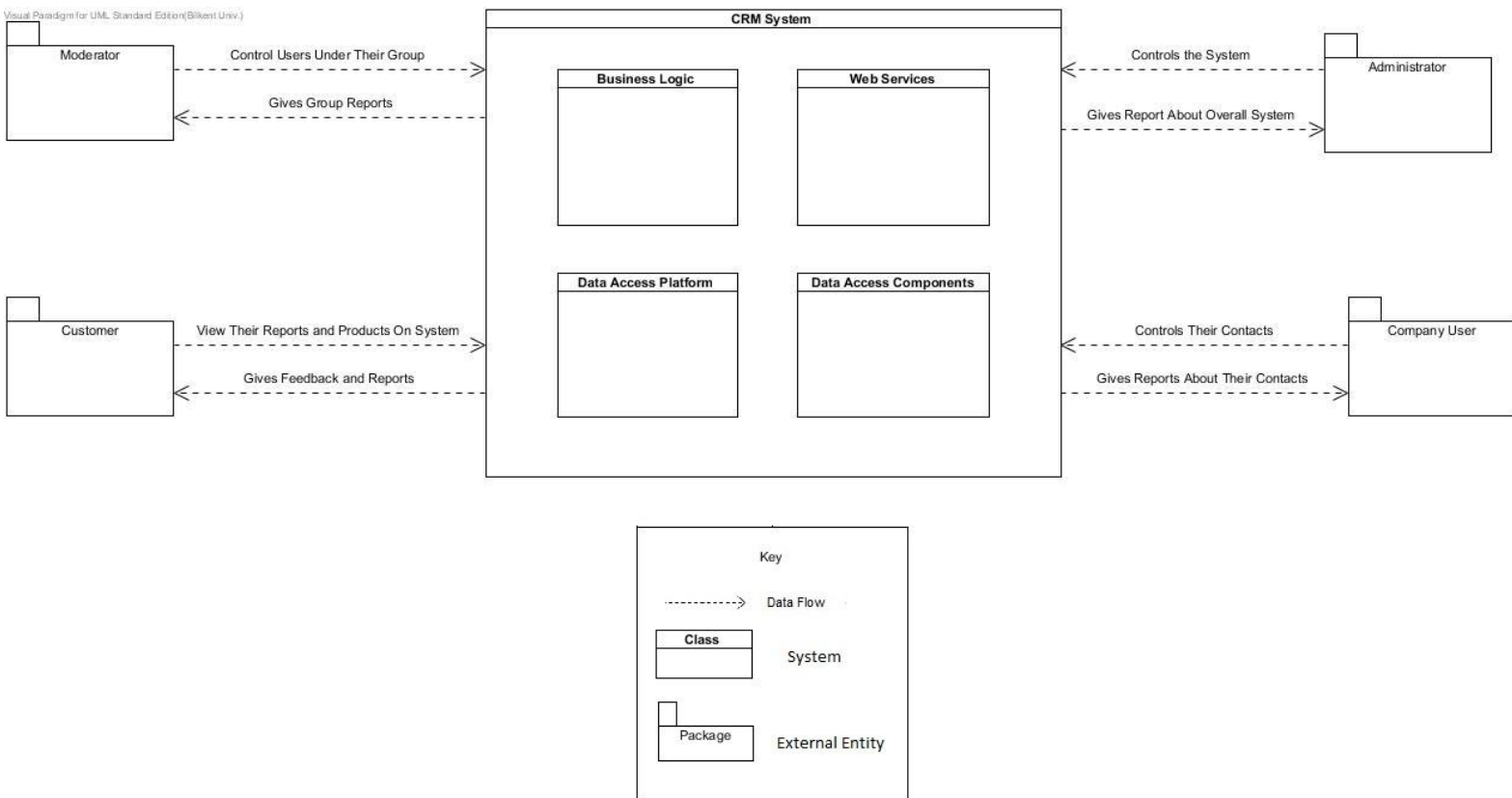


Figure 10 Top Level Context Diagram

Top level context diagram of the system includes four other internal systems which are Business Logic, Web Services, Data Access Platform and Data Access Components. In addition to these, there are four external entities which interact with the system itself. These external entities are:

- Moderator is one of the external entities that interact with the system. System gives reports about moderator's group and moderator controls the users under their group through the system functions.
- Customer is also an external entity who views related reports and products through system. They get feedback and report from the system.
- Administrator controls the overall system as a external entity and system gives reports about overall system to administrator.
- Company User is also an external entity that controls their contacts and gets reports from the system about their contacts and so on.

9. Module Views

9.1. Decomposition View

9.1.1. Decomposition View Context Diagram

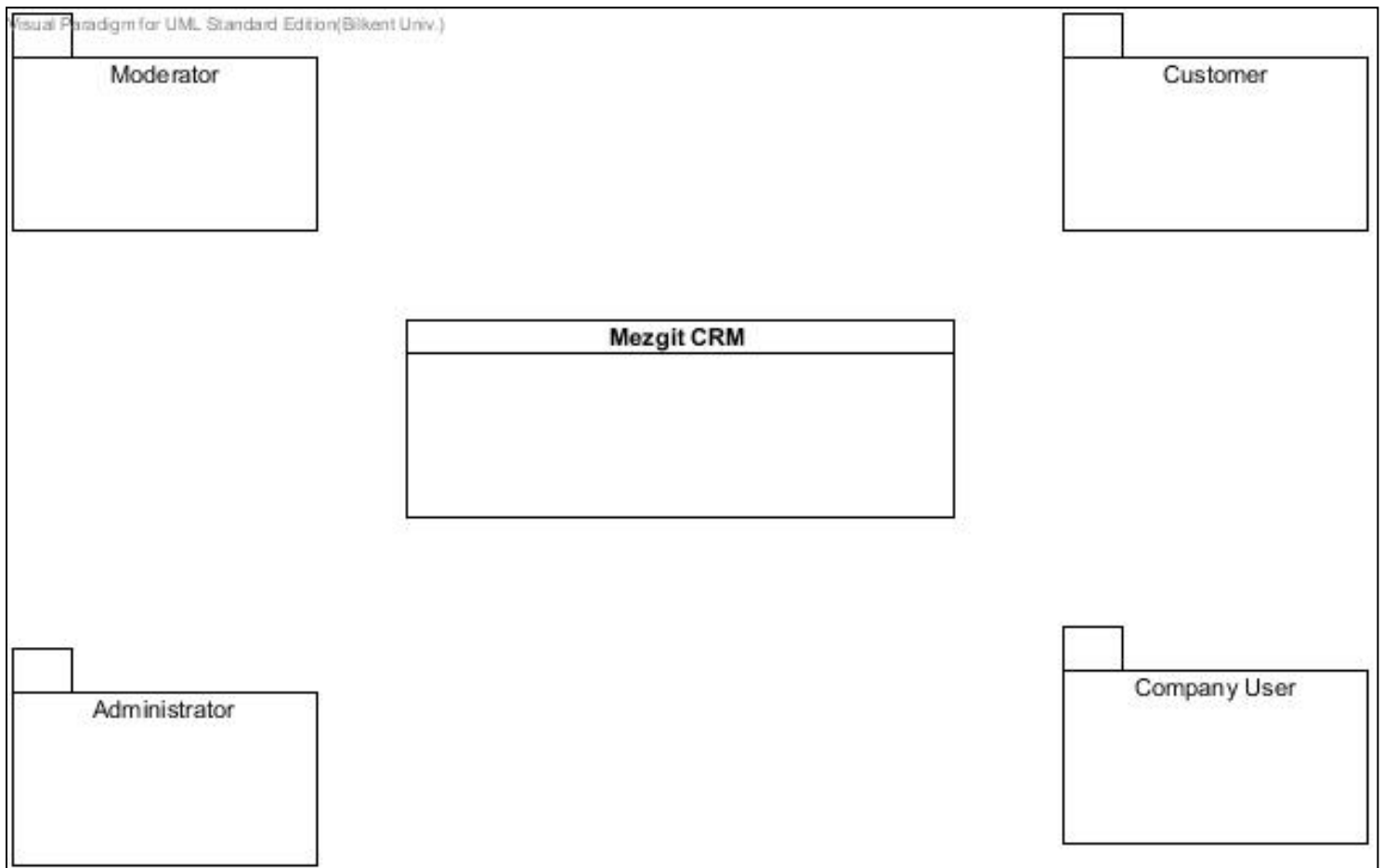


Figure 11 Decomposition View Context Diagram

Decomposition view context diagram shows the environment of the Mezgit CRM system. Mezgit CRM has interactions with Customer, Company User, Moderator and Administrator. Following diagram shows the decomposition view of the context diagram independent from the environment.

9.1.2. Decomposition View for Mezgit CRM

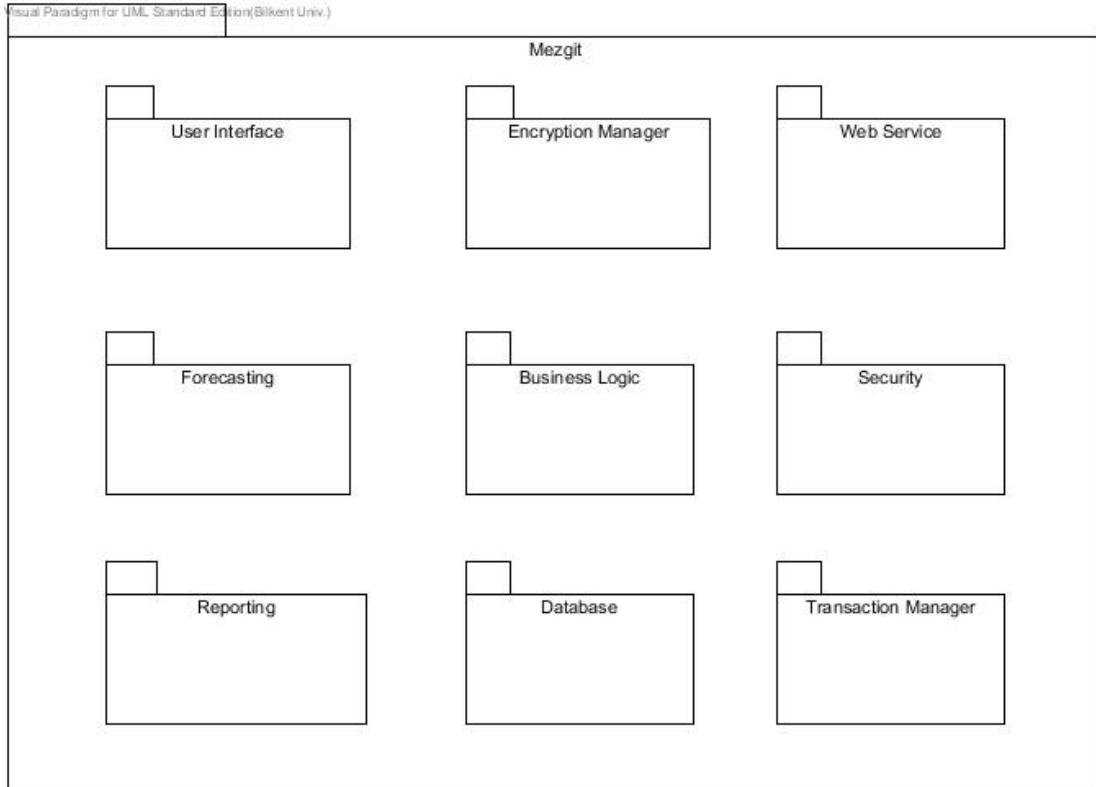


Figure 12 Decomposition View

Decomposition view shows the is-part of relationship between the modules. Main motivation to use decomposition view is to apply divide and conquer approach to the architecture. In the decomposition view of the Mezgit, there are 9 different modules. The relationship between modules (i.e. uses, generalization ...) is not the concern of the decomposition view but roles of the individual modules are.

User Interface Module is the module that provides the user interface to the system. Encryption Module is designed to encrypt the data to provide privacy. Role of the web services is to provide communication between the underlying platform layers.

Business Logic is the main module is the system, that process data, manages the other modules and provides the communication between database and the web services. Forecasting module is the main data processor which analyzes existing data and forecasts about the future products/services of the system with respect to collected data. Reporting module reports the data taken from business logic and forecast manager.

Security is used to provide security to avoid unauthorized accesses. Transaction manager manages the transaction to improve performance and database module is the main database of the system.

9.2. Uses View

9.2.1. Uses View Context Diagram

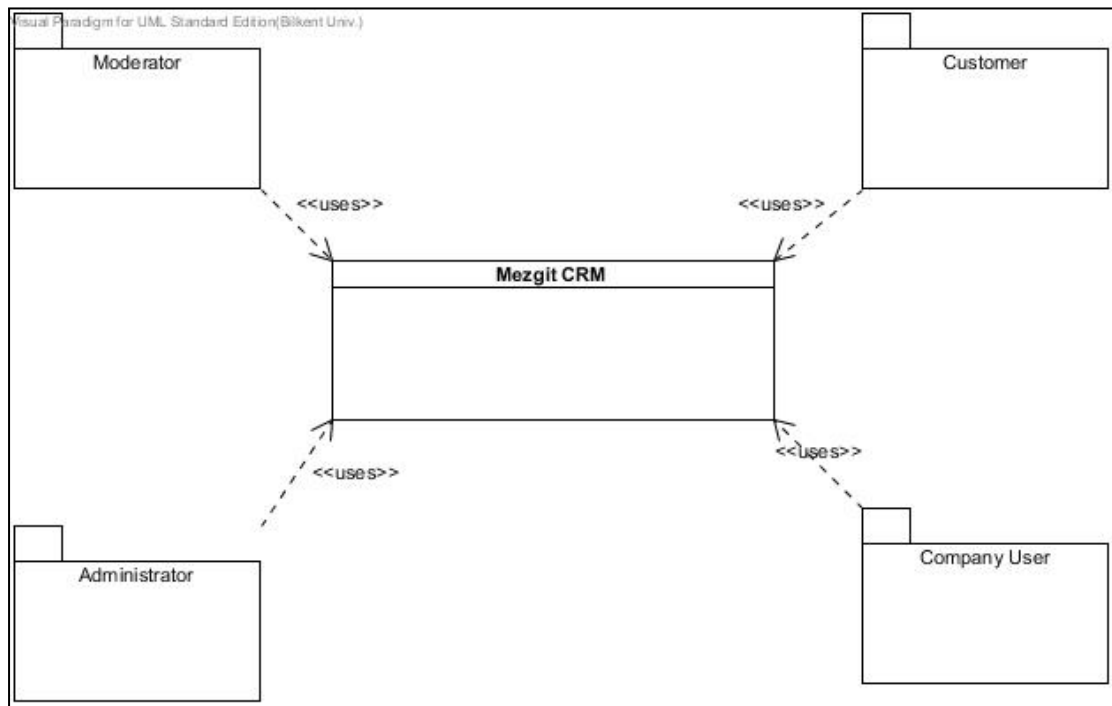


Figure 13 Context Diagram for Uses View

Context diagram for the uses view shows the uses relationship of the system with the environment. Since the external entities are the users, Mezgit CRM does not use any external entity but they use it. With this relationship, Mezgit interacts with the environment.

9.2.2. Uses View for Mezgit CRM

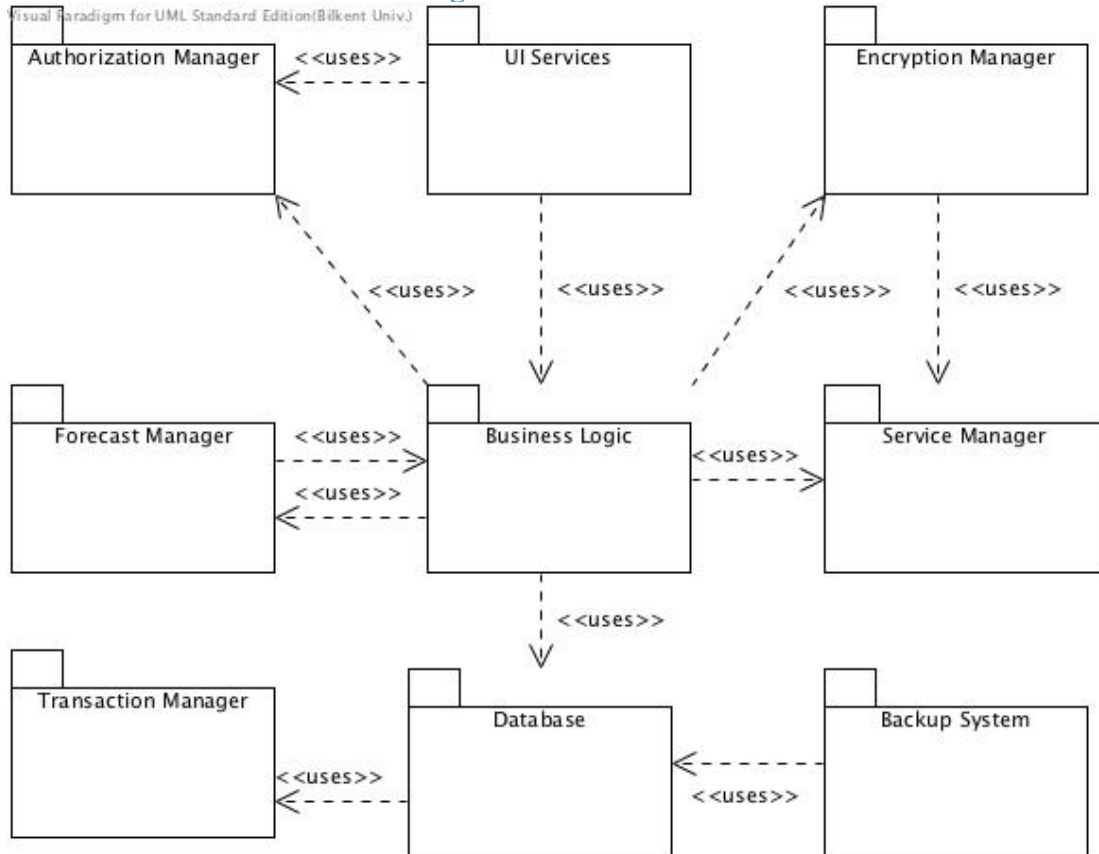


Figure 14 Uses View

The uses style shows the relationships between modules and sub-modules. This view is very helpful for implementing, integrating and testing the system.

User Interface Services are related only interface. However, interface is not stable in Mezgit. UI changes depending on the user type. Therefore it uses Authorization Manager to change UI according to the user type. It also uses Business Logic to show business related context in UI. Business Logic plays a vital role in Mezgit. Therefore, it uses forecast manager to evaluate data for future forecasting. It also uses encryption manager to encrypt data that are instant messaging and emailing. Business logic also uses database. It evaluates the data and provides it to other services. Backup system uses database in order to satisfy crash recovery. Database manager uses transaction manager because in Mezgit, there are significant amount of users. To handle users request stable transaction manager is necessary.

Table 19 Uses View Matrix Representation

	Authorization Manager	UI Services	Encryption Manager	Forecast Manager	Business Logic	Service Manager	Transaction Manager	Database	Backup System
Authorization Manager		0	0	0	0	0	0	0	0
UI Services	1		0	0	1	0	0	0	0
Encryption Manager	0	0		0	0	1	0	0	0
Forecast Manager	0	0	0		1	0	0	0	0
Business Logic	1	0	1	1		1	0	1	0
Service Manager	0	0	0	0	0		0	0	0
Transaction Manager	0	0	0	0	0	0		0	0
Database	0	0	0	0	0	0	1		0
Backup System	0	0	0	0	0	0	0	1	

Design structure matrix also shows the uses relations among modules. Note that key “1” means module in row uses module in column.

9.3. Generalization View

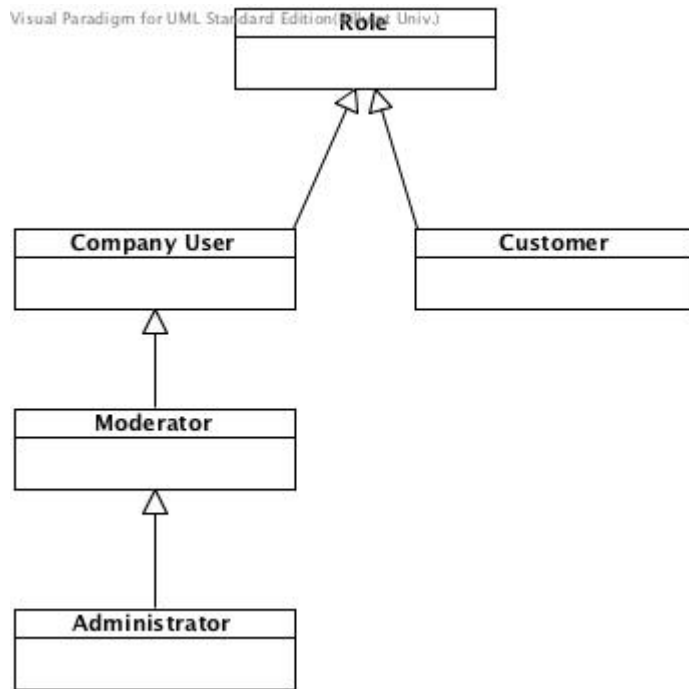


Figure 15 Generalization View

Generalization view is used to show is-a relation between the entities. In other words we can define which entity inherits from the other one.

Role is the entity showing the users of the system. There are 4 different roles at the Mezgit, Company User, Customer, Moderator and Administrator. Within these roles, Company user and Customer have the least significance and also they have the most limited authorization. But Moderator, which generalizes the Company User can perform every action that Company User can and also more. Likely, Administrator can perform every action that Moderator can do and more. Therefore there are the relationships Company User inherits from Role, Moderator inherits from Company User and Administrator inherits from Moderator.

9.4. Layer View and Context Diagram for Layer View

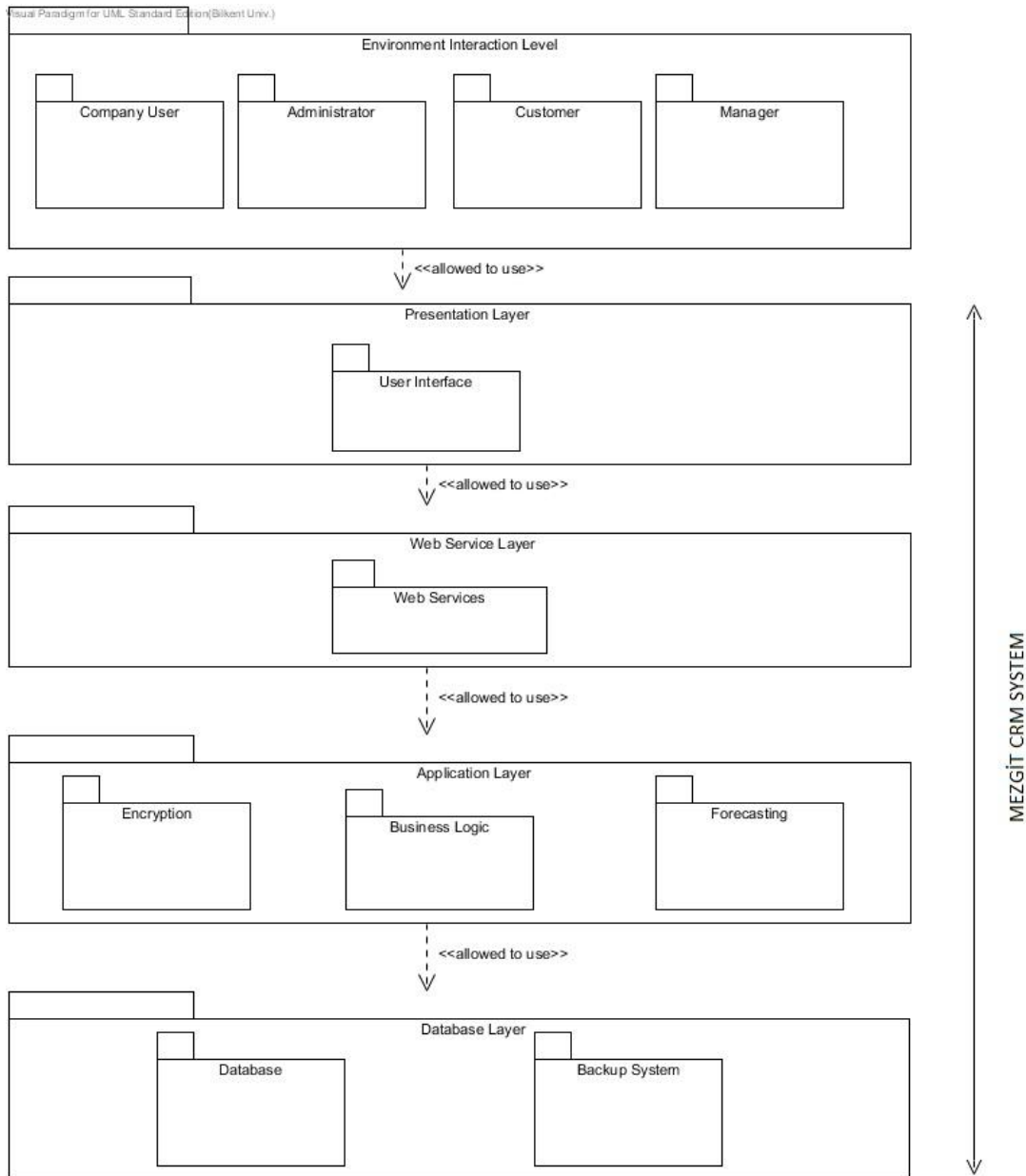


Figure 16 Layer View and Context Diagram for Layer View

Layered views is used to show allowed to use relationship between the layers. Layer is a collection of the software units, modules in the Mezgit.

Figure 16 Layer View and Context Diagram for Layer View show the layer view of the Mezgit CRM and also environment. Since the only interaction with the

environment is with the users and users can only interact with presentation view, environment is at the top of the top level layer of the Mezgit CRM system.

Mezgit project has 4 different layers. Top most one is the user interface layer, none of the other layers can use User Interface layer but it can use the Web Service Layer. Web Service layer contains web services logic and it can use the Application Layer. Application Layer contains the Encryption Manager module, Business Logic module and Forecasting Manager module. It uses the database layer. Down-most layer is the database layer which does not use any layer but used by Application Layer.

9.5. Aspect View

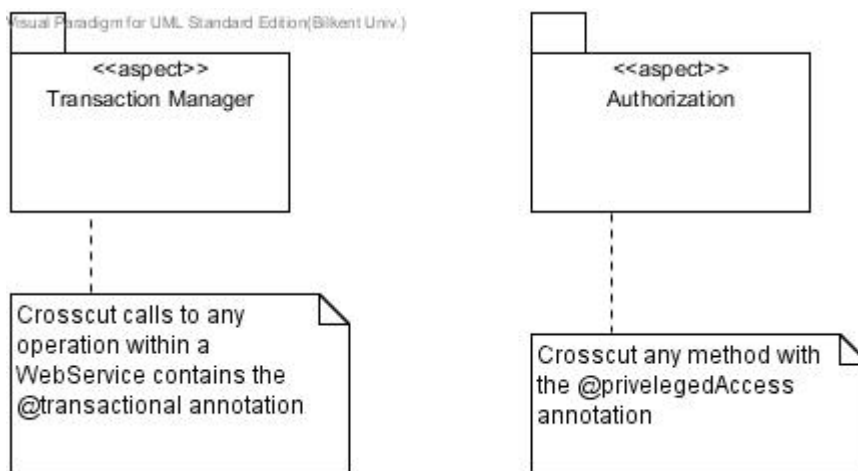


Figure 17 Aspect View

Aspect View is used for showing crosscutting of the system. Mezgit contains two aspects one of them is Transaction Manager which crosscut for any operation within a web service @transactional annotation and the other one is Authorization which is crosscut for any method with the @privelegedAccess annotation.

- Moderators are controlled by Administrator.
- Administrator can create other Company Users.
- Company Users of the system can create Documents.
- Company Users can create Opportunities.
- Users have Documents associated with them and they can edit these documents.

These are the relations in our system. Type of the relationships such as one to many, many to many can be seen on the figure itself.

10. Component and Connector Styles

10.1. Blackboard View

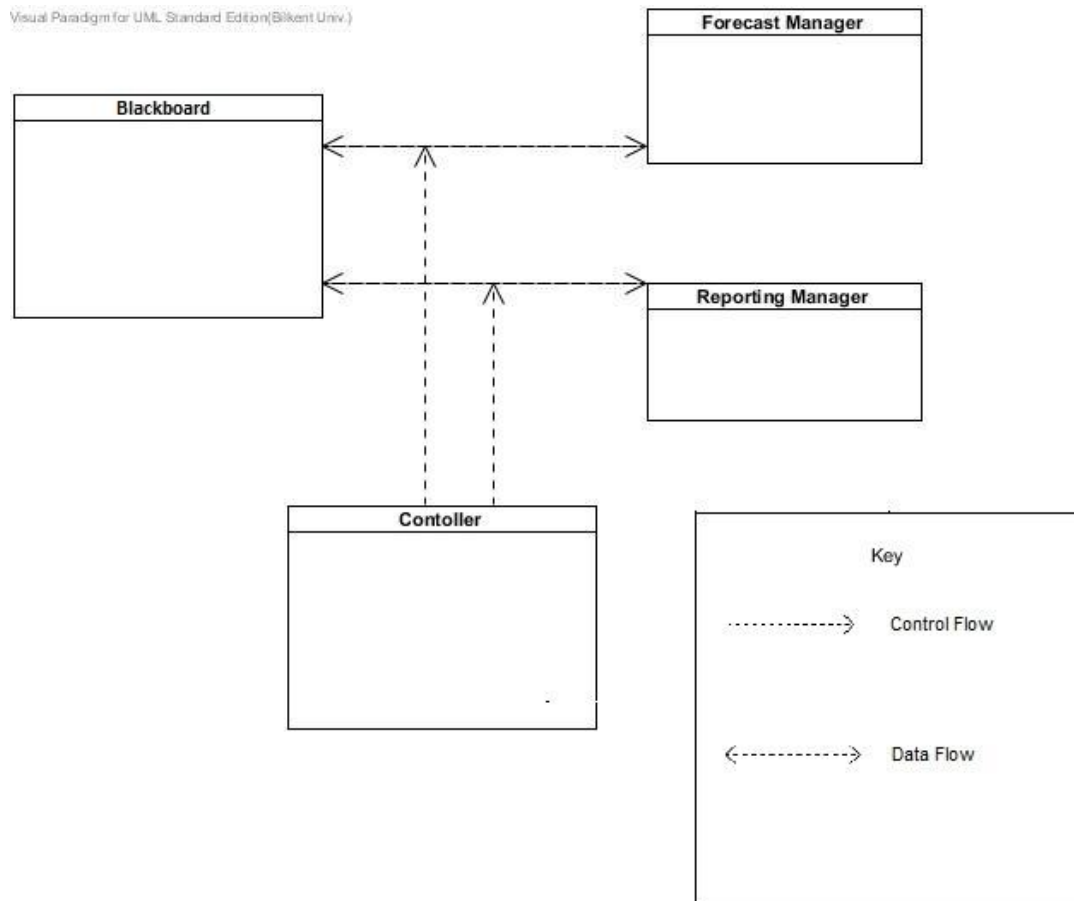


Figure 19 Blackboard View

Central data is stored in our database and it is the Blackboard component of the view. Two knowledge sources Forecast Manager and Reporting Manager gets data from Blackboard and solves different problems. Forecast Manager uses the existing data and information for making future forecasts for different problems. Reporting Manager uses data that creates reports and results from existing data such as sales and marketing information. As a result of their process, they update the blackboard. While these are happening our controller component constantly monitor blackboard.

10.2. Three Tier View

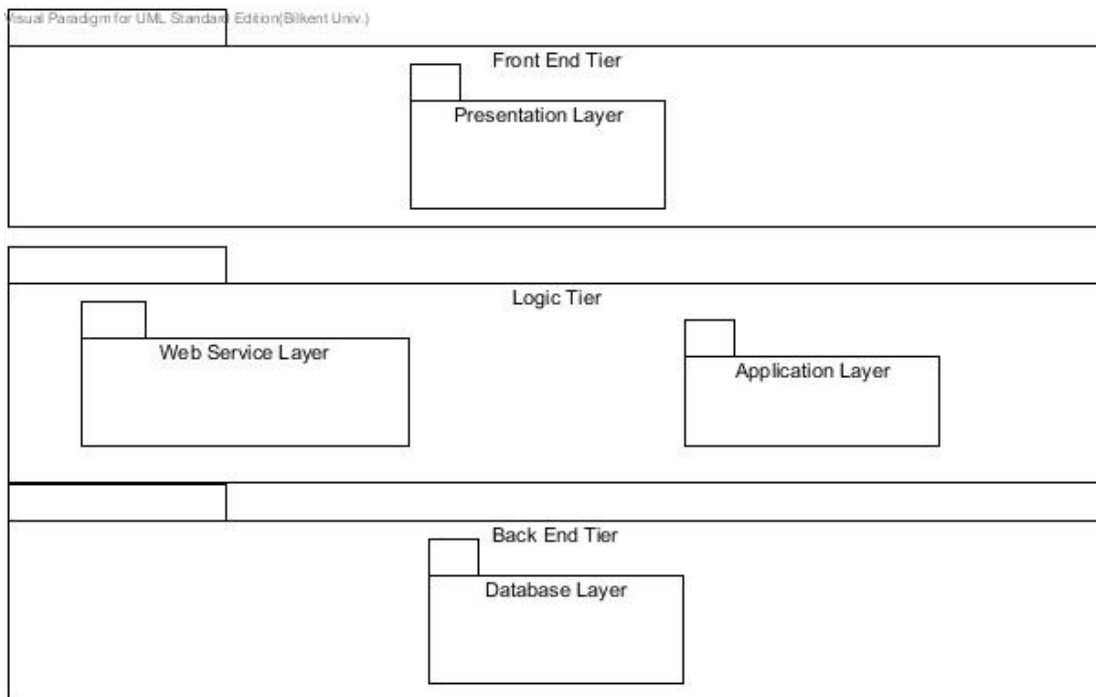


Figure 20 Three Tier View

Mezgit CRM is a project that runs as a web application. At the most of the web applications, three tier approach is applied. Therefore, in the architectural document of this project, three tier view is required. Mezgit CRM has three tiers, front-end, logic and back-end. Since the layers can be mapped to the tier, we have shown that view with the layers as well. Front-end tier contains presentation layer which is user interface services. Logic tier contains web service and application layer and back-end layer contains database layer.

viewpoints and features of the cloud computing, we have defined a new candidate viewpoint which is named as cloud view.

Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a metered service over a network. [4] To describe with the existing systems, we can say that every client at the system are publishers and subscribers as well. Therefore, we can say that cloud computing is a combination of peer to peer approach and also publish-subscribe approach. Figure 22 Cloud View shows the cloud-computing instance that named Azure and the client machine. Relation between these two is publishing and subscribing. Client machine can interact with the web front-end and this interaction stored at the cloud storage.

11. Allocation Views

11.1. Deployment View

11.1.1. Deployment View Context Diagram

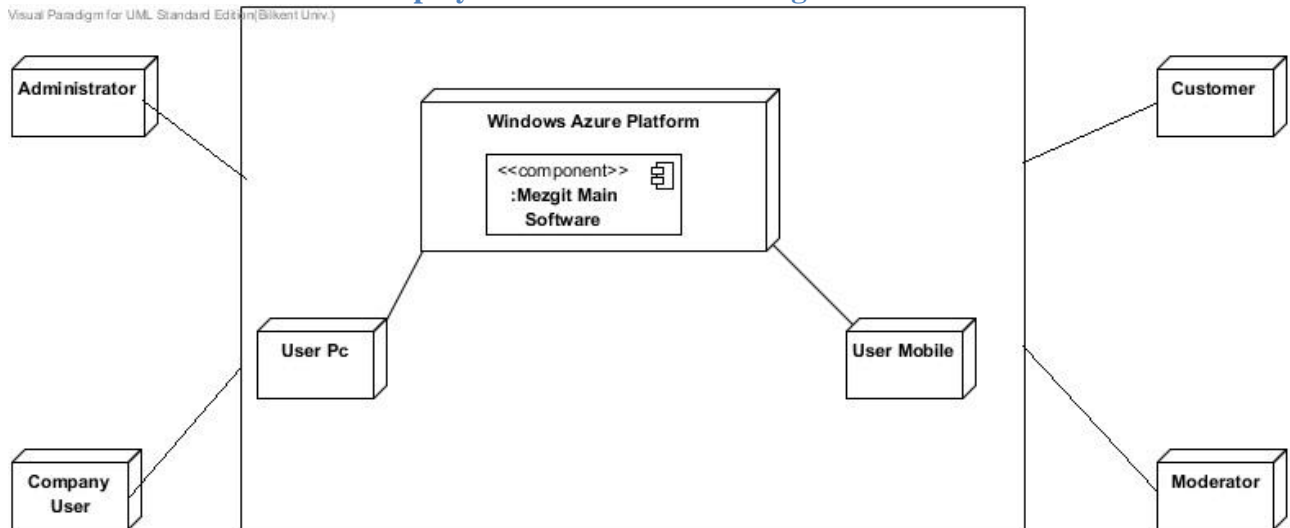


Figure 23 Context Diagram for Deployment View

Context diagram for deployment view shows the external hardware to the Mezigit system. Normally users' personal computers and mobile devices are taken into account but since the hardware of the different user type are not relevant to the implementation of the Mezigit CRM, it has shown as external nodes.

11.1.2. Deployment View for Mezigit CRM

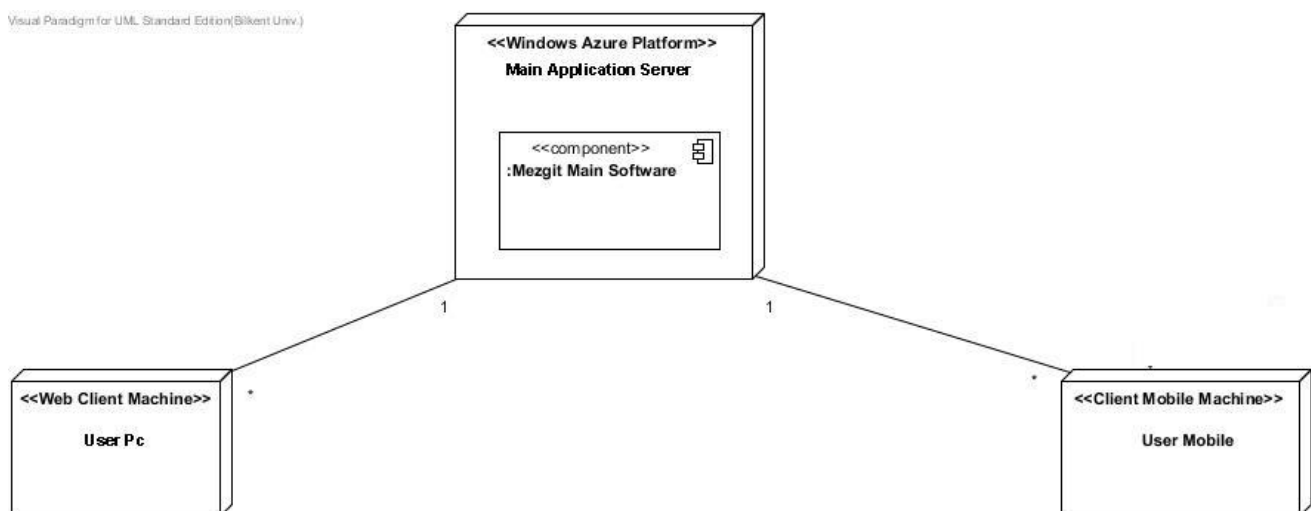


Figure 24 Deployment View

Mezigit CRM consists of three hardware nodes which are Microsoft Azure Platform, Web Client Machine and Client Mobile Machine. Our main application and all

important modules run on Microsoft Azure Platform. This platform provides both a platform to run application and storage the store data. In addition to these, end users need to use computers the access the functionality of program. However, since our application is hosted on cloud, no software needs to be installed on user machine. Any web browser would be sufficient to access application. Lastly, user mobile devices can be used to access application.

11.2. Install View

Visual Paradigm for UML, Standard Edition (Bilkent Univ.)

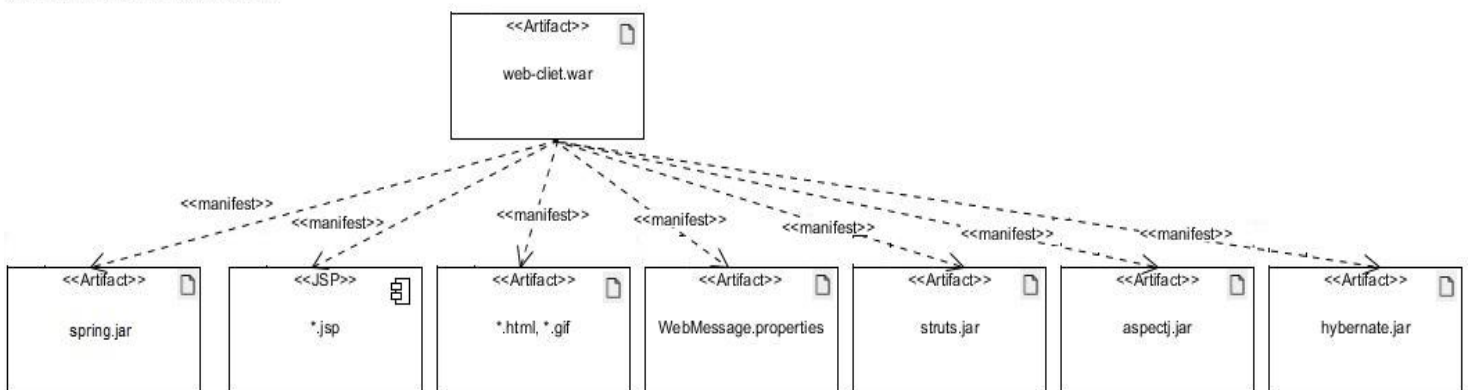


Figure 25 Install View

Mezgit CRM is a web application and will be developed in Java Programming Language. Our install view has a Java Web Archive file that manifests other Artifacts.

- External frameworks will be used during the development. Therefore, additional .jar files will be included in final install view.
- .html files and other images together can be considered as separate artifact.

11.3. Work Allocation View

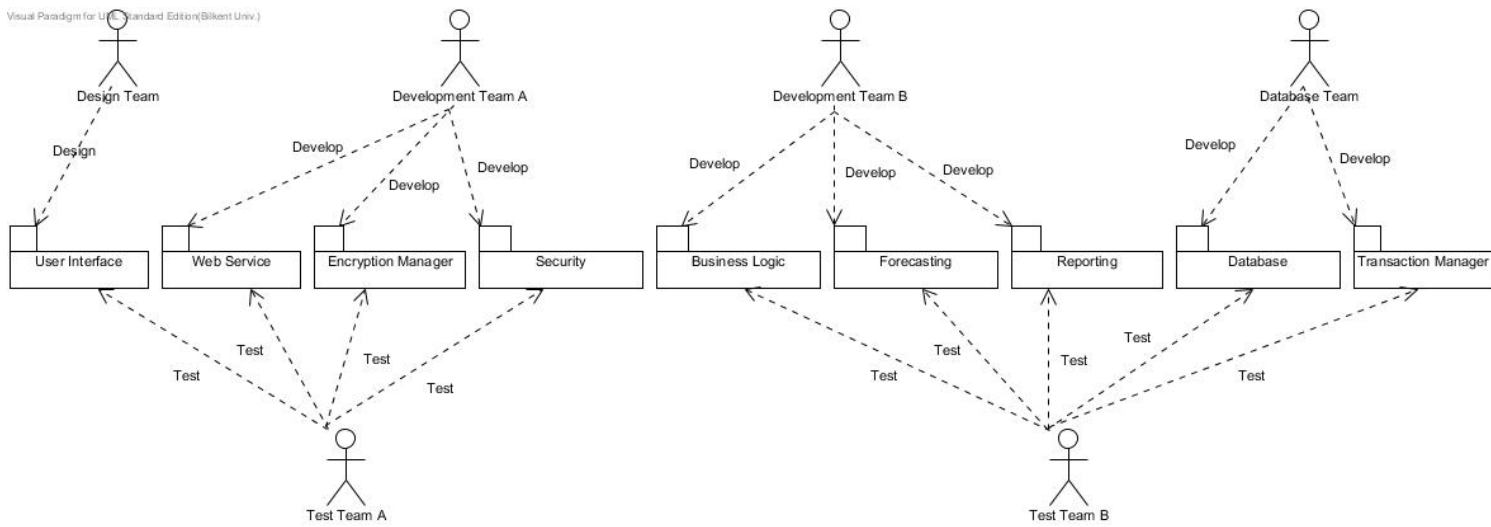


Figure 26 Work Allocation View

There are four development teams and two test teams allocated to work on Mezgit CRM project.

- User Interface will be designed by Design Team.
- Web Service modules, Encryption Manager and Security modules will be developed by Development Team B.
- Data related modules; Database and Transaction Manager will be developed by Database Team.
- Business Logic, Forecasting and Reporting modules will be developed by Development Team C.
- User Interface, Web Services, Encryption Manager, Security will be tested by Test Team A.
- Business Logic, Forecasting, Reporting, Database and Transaction Manager will be tested by Test Team B.

12. Software Architecture Evaluation

In order to evaluate Mezgit CRM architecture, scenario-based analysis evaluation techniques are used. Mezgit CRM architecture are evaluated with scenario-based architecture analysis method and architecture trade-off analysis method.

12.1. Scenario Based Architecture Analysis Method

To assess the impact of anticipated future changes, scenario based architecture analysis method is used.

12.1.1. Description of the Candidate Architecture

Candidate architecture of Mezgit CRM is described below.

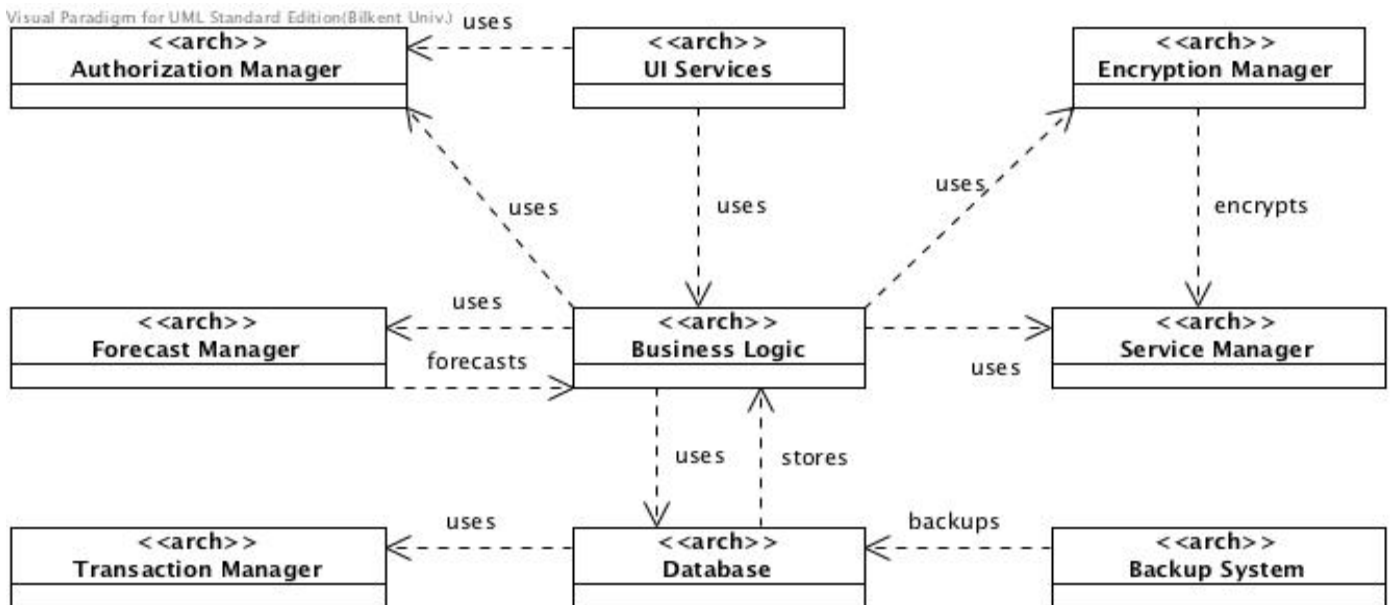


Figure 27 Candidate Architecture

Figure 27 Candidate Architecture shows the architecture of the system with respect to its domain model.

12.1.2. Development and Prioritization of Scenarios

Priority	Scenario	Stakeholder
1	S1. Create a new bid starting on Jan 2012	Company User
2	S2. Change existing groups permissions	System Administrator
3	S3. Consistent UI on different web browsers	Maintainer
4	S4. Avoid unnecessary http request	Maintainer
5	S5. Importing an external contact list to the system	End User, Company User
6	S6. Export existing contacts to other products such as Outlook.	End User, Company User
7	S7. Make minor modifications to the user interface	End User, Company User
8	S8. Generate reports and forecasts documents that support Spreadsheet and Word	End User, Company User
9	S9. Monitor activities of user	System Administrator
10	S10. Port system to mobile-based devices	Maintainer
11	S11. System recovery	Maintainer
12	S12. System availability 24/7	System Engineer, End User

Table 20 Developments and Prioritization of Scenarios

Table 20 captures the scenarios that system must support according to their priorities. These scenarios represent tasks relevant to all stakeholders.

12.1.3. Scenario Evolution

Scenario	Description	Direct/ Indirect	Change Required
1	Create a new bid starting on Jan 2012	Direct	
2	Change an existing groups permissions	Indirect	Database tables, Business Logic and Authorization Manager components must be modified for supporting evolution.
3	Consistent UI on different web browsers	Indirect	UI Services must be reorganized and covered additional web browser support
4	Avoid unnecessary http requests	Indirect	AJAX support must be added to UI Services
5	Importing external contact list to the system	Indirect	This requires to changes to Business Logic
6	Export existing contacts to other products such as Outlook	Direct	
7	Make minor modifications to the user interface	Direct	
8	Generate reports and forecasts documents that support Spreadsheet and Word	Direct	
9	Monitor activities of user	Indirect	This requires to changes to Database and UI Services
10	Port system to mobile-based devices	Direct	
11	System recovery	Direct	
12	System availability 24/7	Direct	

Table 21 Scenario Evolution

12.1.4. Reveal Scenario Interactions

Module	Number of Changes
Business Logic	2
Authorization Manager	1
UI Services	3
Database	2
Forecast Manager	0
Service Manager	0
Backup Manager	0

Table 22 Reveal Scenario Interactions Part 1

Component	Direct Scenarios	Indirect Scenarios
Business Logic	S1, S6	S2
Authorization Manager		S2
UI Services	S7, S10	S9, S3, S4, S5, S9
Database	S1, S6, S8, S12	S2, S5, S9
Forecast Manager	S8	S5
Service Manager	S8	S5
Backup Manager	S11	

Table 23 Reveal Scenario Interactions Part 2

As it is seen in tables 22 and 23, semantically related scenarios affect the same architectural components. Scenario 5 and 9 affects the UI Services and Database. Therefore, cohesive component performs only one concern.

12.1.5. Overall Evaluation

Most of the architectural problems are related to UI management in Mezgit CRM. CRM systems are used for lots of users. Therefore, reducing the HTTP requests is very crucial. In addition, to provide a simple system, consistent UI is very important. Therefore UI Services should robust. UI services should also provide allowing users to make minor modifications on UI. Therefore, it should be flexible. To provide all of these, an MVC pattern might be injected to UI Services. There might also some

database related problems may occur. Depending on the scenarios 2,5 and 9 ER diagrams and table structure might change.

12.2. Architecture Trade-off Analysis Method

Architecture trade-off analysis method is used to understand consequences of architectural decision based on multiple attribute requirements. In addition, ATAM is used in order to understand the interaction between those quality attributes.

Quality Attribute	Attribute Refinement	Scenarios
Performance	Transaction response time	An administrator updates a company's information while the system is under peak load, and the transaction completes in less than 0.5 seconds. (H,M)
		An administrator updates a company's information while the system is under peak load, and the transaction completes in less than 0.5 seconds. (L,M)
	Throughput	At peak load, the system is able to complete 400 normalized transactions per second. (H,M)
	Generating reports	A user requests a report from the system which includes forecasting reports for company IBM in March 2013 while the system is under peak load, and report generating completes in less than 1.75 seconds (H,L)
	Data Latency	Reduce storage latency on customer database to less than 200ms (H,H)
Modifiability	New Product Categories	No scenarios suggested
	Change Costs	Change web user interface in < 4 person-weeks (H,L)
Availability	H/W Failure	Restart after disk failure in < 5 minutes (M,M)
		Network failure detected and recovered in < 1.5 minutes (H,M)

Quality Attribute	Attribute Refinement	Scenarios
Security	Data Confidentiality	Customer database authorization works 99.9% of the time (L,H)
		Instant message and e-mail encryptions are secure 99.9% of the time (L,H)
	Data Integrity	No scenarios suggested

Table 24 Utility Table

13. Conclusion

In this report, a Customer Relationship Manager named “Mezgit” has introduced. To do this, firstly, an introduction to CRM system has been written. After these problem of the current situation has mentioned and need for a CRM system explained. After that followed steps of the software architecture design process are explained. As the fourth step requirements of the system explained and with respect to them use cases, scenarios and prototypes are defined. Then, technical problem analysis and with the help of it domain analysis conducted. With the outcome of domain analyses, concepts defined, conceptual software architecture design model drawn.

During these steps, most important lesson has learned is the importance of the technical problem analysis and domain analysis. Without having proper analysis of technical problem, it is understood that having a domain is impossible and after domain analysis, it is realized that redefining technical problem analysis is needed. So that it can be said that, there is an important iteration between technical problem analysis and domain analysis. And after having proper analysis and defining domain concepts, modeling the architecture is a clear process.

In this project, after having collaboration between the project members, we thought that we can go through software architecture design before domain analysis. Then, we realized that is impossible to draw a correct model without domain analysis. Then we started domain analysis but with the leak of the technical problem analysis, we again could not do it. Then after spending long times at the technical problem analysis and domain analysis, we have figured out how to model software architecture.

After modeling conceptual software architecture, starting with the top level concept diagram, module views, component and connector views and allocation views have been defined, After then, by using Scenario Based Architecture Analysis and Architecture Trade-off Analysis Method, we have evaluated our architectural views and defined which changes should made at the architecture with the their prioritizations and also feasibility.

At the end, this report became a complete software architectural document including conceptual architecture, top level context diagram, architectural views and also evaluation of it.

14. References

[1] <http://www.destinationcrm.com/Articles/CRM-News/Daily-News/What-Is-CRM-46033.aspx>

[2]: [crmarchitecture_whitepaperhpc.pdf](#)

[3] <http://www.slideshare.net/penguincrm/do-you-need-crm>

[4] http://en.wikipedia.org/wiki/Cloud_computing