CENG 491
INITIAL DESIGN REPORT

DOCUMENT FLOW MANAGEMENT SYSTEM

ALPER TUNGA GULBAHAR 120302-5
NERGIS TAS 120351-2
SEVCAN KAYACI 120315-7
MEHMET EROGLU 120296-9
TABLE OF CONTENTS

TABLE OF CONTENTS............................................................................................................. 2
1. PROJECT DESCRIPTION........................................................................................................... 3
   1.1 SCOPE ................................................................................................................................. 3
       1.1.1 GOALS AND OBJECTIVES ......................................................................................... 3
       1.1.2 SYSTEM STATEMENT OF THE SCOPE ....................................................................... 3
2. PROJECT ESTIMATION ......................................................................................................... 5
   2.1 FUNCTION POINTS EVALUATION ..................................................................................... 5
   2.2 COCOMO EVALUATION ..................................................................................................... 6
   2.3 FUNCTION POINT BASED APPROXIMATIONS ............................................................... 7
   2.4 LOC BASED APPROXIMATIONS ....................................................................................... 7
3. PROJECT PLAN & SCHEDULING ......................................................................................... 8
   3.1 WORK PACKAGES ........................................................................................................... 8
       WP1: .................................................................................................................................. 8
       WP2: .................................................................................................................................. 8
       WP3: .................................................................................................................................. 8
       WP4: .................................................................................................................................. 8
       WP5: .................................................................................................................................. 8
       WP6: .................................................................................................................................. 8
       WP7: .................................................................................................................................. 9
   3.2 WORK PACKAGE BREAKDOWN ....................................................................................... 9
       WP1: .................................................................................................................................. 9
       WP2: .................................................................................................................................. 9
       WP3: .................................................................................................................................. 9
       WP4: .................................................................................................................................. 9
       WP5: .................................................................................................................................. 9
       WP6: .................................................................................................................................. 9
       WP7: .................................................................................................................................. 9
   3.3 TIME SCHEDULE FOR THE PROJECT DEVELOPMENT ............................................... 10
4. DATA DESIGN ..................................................................................................................... 10
   4.1 ER DIAGRAM .................................................................................................................. 10
   4.2 DATA FLOW DIAGRAMS ................................................................................................. 12
       4.2.1 DATAFLOW DIAGRAM LEVEL 3 ............................................................................... 12
   4.3 DATA DICTIONARY .......................................................................................................... 14
   4.4 CONTROL FLOW DIAGRAM ......................................................................................... 32
       4.4.1 INTERRUPTS: ........................................................................................................... 32
   4.5 STATE TRANSITION DIAGRAM ..................................................................................... 34
5 OBJECT ORIENTED ANALYSIS ....................................................................................... 39
   5.1 CLASS DIAGRAMS ......................................................................................................... 39
   5.2 COLLABORATION DIAGRAM: ......................................................................................... 40
   5.3 SEQUENCE DIAGRAMS .................................................................................................. 42
6. ARCHITECTURAL DESIGN ............................................................................................... 45
   6.1 TRANSFORM MAPPING .................................................................................................. 46
7. INTERFACE DESIGN ........................................................................................................ 55
   7.1 INTERFACE DESIGN RULES .......................................................................................... 55
   7.2 EXAMPLES FOR GUI ..................................................................................................... 57
APPENDIX............................................................................................................................. 59
1. PROJECT DESCRIPTION

1.1 SCOPE

1.1.1 GOALS AND OBJECTIVES
Main purpose of this project is to design a data flow management system which will decrease the negative effects of bureaucracy by avoidance of paper based storage of data in a company. When all data is stored on papers, people using the documents are compelled when they want to access the document or want another version of the document. Even to glance at a document takes a lot of time. Moreover, it may be unsafe to transport a document from one room to another in a company.

Goals of this system are:
✓ To avoid papers by doing every flow via the internet and local network
✓ To ensure a user friendly interface
✓ To make anonymous users which have requests from the company registered
✓ To provide employees of the company have special rights about their tasks that they are responsible from.

1.1.2 SYSTEM STATEMENT OF THE SCOPE

Customer requirements:
✓ Software will provide different document flows.
✓ This document flows are created by the project manager.
✓ These flows will enable documents communicate with the users in a hierarchical way.
✓ If needed, determined users will interrupt the flow and the system.
✓ System will do reporting and preparing statistical data automatically.
✓ This reports and statistics are sent to the related employees automatically.
✓ Reports and data in will be stored in a database interacting with the employees of the company.

✓ System will create daily and weekly report of the employees’ tasks automatically.
✓ We will have two types of users which will be web-based users and local network users. Web-based users are divided into two groups: anonymous users and registered users. Local network users are divided into four groups: project leaders, managers, employees and administrators which also have three different types: employee account admin, security admin, database admin.
✓ The flows processing level can be seen by the selected persons.
✓ Registered user who composes the document and submit this into the flow for his needs will be able to look at the information about the flow of his/her document in a secure way.
✓ System will identify a unique id for the documents that have been put into the database.
✓ System will identify a unique id for the anonymous users.
✓ System will identify a unique id for the registered users.
✓ There will be user interface that will provide the customers, employees, etc. to use the system easily.
✓ Data flow will be fast.
✓ All documents will flow in a secure way.
✓ Specific users will be authorized to interrupt the flow if needed.
✓ System will prepare the available documents for each user according to the user rights.
✓ Different security levels are needed.
✓ There will be seminars for ensuring any employee use the program easily.
✓ There will be an understandable “Help” menu.
✓ There will be documentation.
2. PROJECT ESTIMATION

2.1 FUNCTION POINTS EVALUATION

Function point estimation focuses on information domain values rather than software functions.

<table>
<thead>
<tr>
<th>Measurement Parameter</th>
<th>Weighting Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Count</td>
</tr>
<tr>
<td>Number of Inputs</td>
<td>19</td>
</tr>
<tr>
<td>Number of User Outputs</td>
<td>71</td>
</tr>
<tr>
<td>Number of user inquiries</td>
<td>16</td>
</tr>
<tr>
<td>Number of Files</td>
<td>23</td>
</tr>
<tr>
<td>Number of External Interfaces</td>
<td>5</td>
</tr>
</tbody>
</table>

Count Total…………………………………………………………………………………….. 735

Note: a *b implements that there a counts of b weighting factor related with that information domain value.

Complexity Adjustment Values

1. Does the system require reliable backup and recovery? (3)
2. Are data communications required? (5)
3. Are there distributed processing functions? (4)
4. Is performance critical? (4)
5. Will the system run in an existing, heavily utilized operational environment? (5)
6. Does the system require on-line data entry? (5)
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations? (2)
8. Are the master files updated on-line? (1)
9. Are the inputs, outputs, files or inquiries complex? (4)
10. Is the internal processing complex? (4)
11. Is the code designed to be reusable? (4)
12. Are conversion and installation included in the design? (4)
13. Is the system designed for multiple installations in different organizations? (4)
14. Is the application designed to facilitate change and ease of use by the user? (5)
Function points are evaluated using this formula:

\[ FP = \text{count\_total} \times (0.65 + 0.01\times\sum(F_i)) \]

where \( F_i \) is the complexity adjustment value for each answer \((0 \leq F_i \leq 5)\)

\[ FP = 735 \times (0.65 + 0.01 \times 54) \]
\[ FP = 874.65 \]

### 2.2 COCOMO EVALUATION

COCOMO is one of the most widely used and discussed software cost estimation models. LOC value will be calculated from FP (calculated in section 5.1) to do the COCOMO calculation. To find LOC and effort we have to choose the programming language also. We will use .NET. LOC/FP for programming languages which we will use in .NET is approximately 32. This value is found from approximating LOC/FP’s for programming languages such as java (30 LOC/FP on average), c* (32 LOC/FP on average).

\[ \text{LOC} = \text{FP} \times 32 \]
\[ \text{LOC} = 874.65 \times 32 = 27989 \text{ LOC} \approx 28 \text{ KLOC} \]

According to COCOMO, effort \( E \) and duration \( D \) are

\[ E = a_i \times \text{KLOC}^b_i \times \text{EAF} \]
\[ D = c \times E^d \]

where \( a, b, c \) and \( d \) are constants depending on your project type. We consider that our project will be semi-detached (medium). For our project, EAF (Cost Driver) is approximately 1.01 (after calculations).

\[ E = 3.0 \times (28)^{1.12} \times 1.01 \]
\[ = 126.55 \text{ person-month} \]
\[ D = 2.5 \times (126.55)^{0.35} = 13.61 \text{ months} \]

Average staffing = (126.55 Person-Months) / (13.61 Months) \( \approx 9 \) people

**COMMENT:** In our project, we have 4 people in our team. According to calculations from COCOMO2, we found that we need 9 people to finish the whole project in 13.6 months. This refers to 9 months to finish the project with 13 people and 31.6 months to
finish the whole project with 4 people. We believe that when we work hard, we can finish this project in 6 months (from now on).

2.3 FUNCTION POINT BASED APPROXIMATIONS

This part is to use some other estimation techniques on the same FP (FP=874.65) information and to produce a better approximation.

*Albrecht and Gaffney Model:* 
\[ E = -13.39 + 0.0545 \times FP \]
\[ = 34.28 \text{ person-month} \]

*Kemerer Model:* 
\[ E = 60.62 \times 7.728 \times 10^{-8} \times FP^3 \]
\[ = 3134.63 \text{ person-months} \]

*Matson, Barnet, Mellichamp Model:* 
\[ E = 585.7 + 15.12 \times FP \]
\[ = 13810.41 \text{ person-month} \]

These results are very far from the effort evaluated in the COCOMO estimation.

2.4 LOC BASED APPROXIMATIONS

We used two LOC based estimation techniques to produce a better approximation.

*Walston-Felix Model:* 
\[ E = 5.2 \times (\text{KLOC})^{0.91} \]
\[ = 107.87 \text{ person-months} \]

*Boehm Simple Model:* 
\[ E = 3.2 \times (\text{KLOC})^{1.05} \]
\[ = 105.84 \text{ person-month} \]

*Doty Model:* 
\[ E = 5.288 \times (\text{KLOC})^{1.047} \]
\[ = 173.17 \text{ person-month} \]

All results are near to COCOMO estimation when compared with FP based estimations. We can use these results for further estimations.
3. PROJECT PLAN & SCHEDULING

3.1 WORK PACKAGES

WP1:

Project Start
Survey for the current and past document flow management systems
Search for technical details (environment)

WP2:

Software requirement analysis: Requirements are defined, Scenario is developed and process modeling tools are studied. Analysis of scenario and the conceptual model is prepared.

WP3:

Software Design: The software is designed by considering module relationships, data flow and state transitions. The data, architecture, interface and procedures are designed.

WP4:

Software Prototype Development: Requirements are gathered for interface prototype. Case tools are used to build interface prototype.
Software Detailed Design: All detailed data (functions, attributes, dictionaries, etc.) are constructed in addition to the WP3. We work on detailed design while initial design report preparing is in progress.

WP5:

Software Development: The designed procedures are coded with pre specified programming languages using reusable code and development tools on defined systems.

WP6:

Test & Validation: The program modules are tested and validated among design considerations.
WP7:

*Documentation:* Help for the user of the system is prepared. Essential documents are gathered.

### 3.2 WORK PACKAGE BREAKDOWN

**WP1:**

Task 1: Surveying functioning principles of CASE and process modeling.
Task 2: Search on customer needs
Task 3: Getting familiar with .NET, XML, SSL, SQL SERVER 2000 and other related tools

**WP2:**

Task 4: Scenario creation and analysis
Task 5: Preparation of conceptual model
Task 6: Writing requirement analysis report

**WP3:**

Task 7: Describing the objects, determining the attributes and methods for the objects.
Task 8: Writing the initial design report

**WP4:**

Task 9: Building Interface prototype
Task 10: Constructing the detailed design report

**WP5:**

Task 11: Coding the whole project

**WP6:**

Task 12: Testing

*Note:* Testing will also be done after finishing each general part in WP5.
WP7:

Task 13: Preparing help for the users
Task 14: Documentation of all essential information

3.3 TIME SCHEDULE FOR THE PROJECT DEVELOPMENT

This table can be found in the Appendix.

4. DATA DESIGN

4.1 ER DIAGRAM

We can explain our ER diagram according to tables in the databases.

*Customer Personal Data* is the table to store personal data. This table has relationship with two tables: *Customer Login Data* and *Customer Mail*. *Customer Login Data* is used to store the login data of the registered user and *Customer Mail* for keeping the track of the mails that are received by the registered user. *Customer Personal Data* creates GUI for customer mails. Also it creates GUI for customer after ensuring that the registered user proves his/her login data. There is also need for customer mails to be stored for a while. *Customer History Mail Data* is used in interaction with customer mail to do this operation.

*Employee Personal Data* table is used to store personal information of all types of employees (admin, manager, project leader, and employee). Like the customer personal data, it has relations with *Employee Login Data* and *Employee Mail*. *Employee History Mail Data* is used to store read employee mails. Since the employees will be assigned some labor by the project leader the system has to keep track of the assignment data of each employee. *Employee Assignment Data* information and *Employee Personal Data* will be in interaction with each other to create GUI for the employee web page. *User Defined Project* and *PreDefined Project* will also be able to reach *Employee Assignment Data* to ensure which employees are related with the projects. *Employee Personal Data* can reach *Assigned Project* which stores employee-project relation. Thus, *Assigned Project* is in interaction with the *User Defined Project*.

*PreDefined Project* is a simple template project containing *PreDefined Request, PreDefined Response* and *PreDefined Flowchart* assigning some information e.g. levels of flow, identification numbers for predefined requests, responses and flowcharts. They
contain PreDefined Document which shows the already defined document types like word, text, picture. PreDefined Project has definition in PreDefined PDefinition which automatically gives permission to the user to perform operations in the system.

User Defined Project is created after project leader has created the project by inserting some prequest and presponse instances and flowcharts. These are in User Defined PRequest, User Defined PResponse and User Defined Flowchart. They contain User Defined Document which shows updated predefined document types. It has definition in User Defined PDefinition in which the project leader gives permission to the user to perform operations in the system.

Since we have a very big database, some reports and statistics should be available to users for the flow of the project and the documents. There are Employee Reports, User Defined Reports and System Defined Reports where the first is to store report for the performance and works of the employees, the second for reports created by the users and the last for the reports created by the system for during the flow of the project automatically. All these three tables are related with User Defined Project since they are stored during the project. Moreover, statistics are used during the project. System Defined Statistics are one of them. This table creates GUI for the customer and employees. Hence, it is also related with Employee Statistics and Customer Statistics.

When a project is activated, there are tables to store documents of the project process. Active Document and Document Information are used for this purpose. The former stores level of flows and related employees during the project. Thus, Active Document is related with Employee Assignment Data to get information about which employee approved which document. It is also related with the Current Request of the customer database to give information about the level of the customer’s document. Document Information is used to communicate employees and customers with the project. So, it is related with Customer Personal Data, User Defined Project and Active Document.

(For ER Diagram See Appendix)
4.2 DATA FLOW DIAGRAMS

4.2.1 DATAFLOW DIAGRAM LEVEL 3

We have two logins in our diagram. One of them is for web users (customers) and other is for network users (employees).

For the first type of users we have a login screen that compares the given inputs with the data in the database. After authentication the customer logs into the part of the system for customers personal usage. After this action customer has the opportunity to choose between three requests. She can choose secure mailing, information requesting or document generating modules. Secure mailing is described for all users. The specific usages for customers are document operations and Information requests. In document generation phase customer can compose a predefined document or a manual document to start a process. In Information request customer can get the information about her previous processes which she started before via secure mail.

For second type of users there exists also a login screen and same comparisons. After logging into the module user sees her secure mails and respond them. During this respond phase type of user is selected automatically by the system.

In manager module, user has secure mail, project leader assignment, report operation request and project assignment display request and project statistics request. In project leader assignment manager chooses an employee to manage a specific project. With report operation request she can show the progresses of the processes or makes system to create a specific report about the given subject. In project assignment display request shows manager the list of the employees included in the specified project. Project statistic request is for displaying the weekly, monthly or yearly statistics for the current projects.

Another module is Employee Account Admin Module. In this module user can create or update an employee to the current database if he has the status of employee account administrator.
Another admin module is Database Admin Module. Employee deletion, employee insertion, backup databases and report request is main applications for the database administrator. If an employee has database administrator status she can use this module.

Security is on the shoulders of security admin module. A security administrator can assign security levels and request reports.

Last but not least module in our dataflow is project leader module. A project leader can do assignment operations, flowchart operations, document operations and project operations. In assignment operations she can assign employees to the projects she manages. Also she can update the employees assigned. In flowchart operations she can update or create flowcharts defined for document flow. Document operations are for creating or updating documents, prequests or presponses. Lastly Project operations are for creating or updating projects in the database. A user must have the status of Project Leader to use this module.

Our last module is Employee module. An employee have reports request, approve documents, receive or deliver documents in his options.

Secure mailing is a common module for all user types. Via secure mailing a user can read, compose or delete mails. This module makes our system more secure. Moreover we store the deleted mails weekly in our databases to restore in emergency.

(For Dataflow Diagram Level 3 See Appendix)
4.3 DATA DICTIONARY

Name : Customer User Data
alias : none
Where used/how used : Web Login Menu (input)
                         Web-Based User (output)
Description : [username:string|password:string]

Name : Customer Data
alias : none
Where used/how used : Web Login Menu (input)
                         Customer Login Data (output)
Description : [CustomerId:long int|username:string]

Name : Authorization Data
alias : none
Where used/how used : Registered User Menu (input)
                         Web Login Menu (output)
Description : [ValidUser:bool]

Name : Customer Personal Data
alias : none
Where used/how used : Registered User Module (input)
                         Customer Personal Data (output)
Description :
[FldRef:long int|Name:string|Surname:string|Address:space|Telephone:string]
              Telephone = [areacode|local phone number]
              Address = [Street Name|Building Number|Door Number]

Name : User Request
alias : none
Where used/how used : Request Generator( input )
                         Registered User Module( output )
Description : [RequestId:long int|CustomerId:long Int]

Name : Definition Data
alias : none
Where used/how used : Request Generator( input )
                         PDefinition( output )
Description :
[ProjectId:int|ProjectTitle:string|ProjectAlias:string|ProjectAttributes:string]

Name : Generated Request
alias : none
Where used/how used : System Operations( input )
                         Request Generator( output )
Description : [CustomerId:long int|ProjectId:long int]
Name : Document Insertion Data
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Generation Option</td>
<td>[ProjectId:long int</td>
</tr>
<tr>
<td>Document Type Data</td>
<td>[PrequesIds:string]</td>
</tr>
<tr>
<td>Document Attributes</td>
<td>[DocumentId:string]</td>
</tr>
<tr>
<td>Document Data</td>
<td>[RequestText:string</td>
</tr>
<tr>
<td>Loaded Document</td>
<td>[DocType:int</td>
</tr>
<tr>
<td>Process Start Request</td>
<td>none</td>
</tr>
</tbody>
</table>

**Data Types**

- Long Integer: `long int`  
- Boolean: `bool`  
- String: `string`  
- Integer: `int`  
- Float: `float`  

**Where used/how used**

- Document Composer  
- Prepared Document  
- Document Generator  
- Document Composer  
- Project  
- Request  
- Xml Converter  
- Process Start Request  

**Aliases**

- none
Description:

[DocumentId:int|FileId:long int|CustomerId:long int]

Name: Process Data
alias: none
Where used/how used: Process Starter( input )
                         Active |Project( output )

Description:

[FileId:int|ProjectId:int|ApprovalId:bool-Default FALSE-[LevelId:int-Default 1-]

Name: Information Request
alias: none
Where used/how used: Quick Response Module( input )
                      System Operations( output )

Description:

[CustomerId:long int|ProjectId:int]

Name: Customer File Data
alias: none
Where used/how used: Quick Response Module( input )
                      Current Request( output )

Description:

[FileId:long int|FileId:long int|ApprovalDate:date|ApprovedEmployeeId:long int]

ApprovalDate = [day|month|year]

Name: Statistical Data
alias: none
Where used/how used: Quick Response Module( input )
                      Customer Statistics( output )

Description:

[CustomerId:long int|FileId:long int|EmployeeId:long int|ApprovalStatus:bool]

Name: Report Data
alias: none
Where used/how used: Report Display
                      Report Evaluator( input )
                      Quick Response Module
                      Report Evaluator
                      Project Report( output )

Description:

[ProjectId:long int|FileId:long int|EmployeeId:long int|ApprovalStatus:bool]

Name: Mail Request
alias: none
Where used/how used: Employee Account Admin Module
                      Manager Module
                      Database Admin Module
Security Admin Module
Employee Module
Project Leader Module
System Operations (input)
Secure Mail (output)

Description : [UserType:int][UserId:long int]

Name : Compose Request
alias : none
Where used/how used : Mail Composer( input )
                    Secure Mail ( output )
Description : [UserId:long int][UserType:int][MailRequestId:int]

Name : Delete Request
alias : none
Where used/how used : Mail Delete( input )
                      Secure Mail ( output )
Description : [UserId:long int][UserType:int][MailRequestId:int][MailId:int]

Name : Read Request
alias : none
Where used/how used : Mail Read( input )
                      Secure Mail ( output )
Description : [UserId:long int][UserType:int][MailRequestId:int][MailId:int]

Name : Mail Data
alias : none
Where used/how used : Employee Mail
                      Customer Mail
                      Mail Read( input )
                      Employee Mail
                      Mail Delete
                      Mail Composer( output )
Description : [MailId:int][UserId:long int][UserType:int][MailText:string]
MailStatus:bool[SystemMailStatus:bool][Date:date][From:string][To:string]
          Date = [day|month|year]

Name : Employee User Data
alias : none
Where used/how used : Network Login Menu( input )
                      Network User( output )
Description : [Username:string][Password:string]

Name : Account Data
alias : none
Where used/how used : Network Login Menu (input)
                      Employee Login Data(output)
Description :
[EmployeeId:long int|Username:string|Password:string]

Name : Authorization Data
alias : none
Where used/how used : User Type Selector (input)
                       Network Login Menu (output)
Description :
[ValidUser:bool|UserType:int-Default 1-]

Name : Employee Personel Data
alias : none
Where used/how used : User Type Selector (input)
                       Employee Personal Data(output)
Description :
[FieldRef:long int|Name:string|Surname:string|Address:string|
  Telephone:string|StartDate:date|EmployeeStatus:int]
  StartDate = [day|month|year]
  Telephone = [areacode|local phone number]
  Address = [Street Name|Building Number|Door Number]

Name : Employee Assignment Data
alias : none
Where used/how used : User Type Selector (input)
                       Employee Assignment(output)
Description :
[EmployeeId:long int|ProjectId:long int|LevelId:int|IndexId:int]

Name : Employee Data
alias : none
Where used/how used : Employee Module (input)
                       User Type Selector (output)
Description :
[FieldRef:long int|Name:string|Surname:string|Address:string|
  Telephone:string|StartDate:date|EmployeeStatus:int]
  StartDate = [day|month|year]
  Telephone = [areacode|local phone number]
Address = [Street Name][Building Number][Door Number]

Name : Manager Data
alias : none
Where used/how used : Manager Module( input )
                      User Type Selector ( output )
Description : [ManagerId:int]

Name : Report Create Request
alias : none
Where used/how used : Report Creator( input )
                      Manager Module ( output )
Description : [ReportType:int]

Name : Project Report Data
alias : none
Where used/how used : Project Reports( input )
                      Report Creator( output )
Description : [ProjectId:long int|FileId:long int|ApprovalId:int|LevelId:int]

Name : Project Leader Assignment Request
alias : none
Where used/how used : Project Leader Assignment ( input )
                      Manager Module ( output )
Description : [ManagerId:long int|ProjectId:long int]

Name : Project Leader Assignment Data
alias : none
Where used/how used : Employee Assignment( input )
                      Project Leader Assignment( output )
Description : [ManagerId:long int|AssignedPLId:long int|ProjectId:long int]

Name : Project Operations Request
alias : none
Where used/how used : Manager Projects Operations( input )
                      Manager Module( output )
Description : [ProjectId:int|ManagerId:long int]

Name : Project Data
alias : none
Where used/how used : Assigned Project
                      Manager Project Operations( input )
                      Assigned Project
Manager Project Operations (output)

Name: Project Assignment Display Request
alias: none
Where used/how used: Project Assignment Display (input)
Manager Project Operations (output)

Description: [ProjectId:int|ManagerId:long int]

Name: Project Statistics Request
alias: none
Where used/how used: Statistics Operations (input)
Manager Operations Module (output)

Description: [ProjectId:int|ManagerId:long int]

Name: Statistical Request
alias: none
Where used/how used: Statistics Generator (input)
Statistics Operations (output)

Description: [ProjectId:int|ManagerId:long int]

Name: Statistical Data
alias: none
Where used/how used: Statistics Generator (input)
Project Statistics
Employee Statistics (output)

Description:
[ProjectId:long int|EmployeeId:long int|ApprovalDate:date]
CustomerId:long int|FileId:long int]
ApprovalDate = [day|month|year]

Name: Statistics Display Data
alias: none
Where used/how used: Statistics Display (input)
Statistics Generator (output)

Description: [ProjectId:long int|EmployeeId:long int]

Name: Employee Account Admin Data
alias: none
Where used/how used: Employee Account Admin Module (input)
User Type Selector (output)

Description: [AdminId:long int]

Name: New Account Request
alias : none
Where used/how used : Account Creator( input ) 
                      Employee Account Admin Module ( output )
Description :
[AdminId:long int|AccountCreatorRight:bool-Default FALSE-]

Name : Account Personal Data
alias : none
Where used/how used : Employee Personal Data( input ) 
                      Account Creator
                      Account Update( output )
Description :
[EmployeeId:int|FieldRef:long int|Name:string|Surname:string|
Address:string|Telephone:string|EmployeeStatus:int]
      Telephone = [areacode|local phone number]
      Address = [Street Name|Building Number|Door Number]

Name : Account Login Data
alias : none
Where used/how used : Employee Login Data( input ) 
                      Account Creator
                      Account Update ( output )
Description :
[EmployeeId:long int|Username:string|Password:string]

Name : Account Update Request
alias : none
Where used/how used : Account Update Module( input ) 
                      Employee Account Module( output )
Description :
[AdminId:long int|AccountUpdateRight:bool-Default FALSE-]

Name : Database Admin Data
alias : none
Where used/how used : Database Admin Module( input ) 
                      User Type Selector ( output )
Description :
[AdminId:long int]

Name : Database Operation Data
alias : none
Where used/how used : Database Admin Operations ( input ) 
                      Database Admin Module ( output )
Description :
[AdminId:long int|OperationRequest:int|DatabaseAdminOperationsRight:boolean]

Name : Delete Employee Request
alias : none
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert Employee Request</td>
<td>[EmployeeId:long int]</td>
</tr>
<tr>
<td>Employee Data</td>
<td>[EmployeeId:long int</td>
</tr>
<tr>
<td>Backup Request</td>
<td>[BackupRight:bool/AdminId:long int</td>
</tr>
<tr>
<td>Report Send Request</td>
<td>[ReportType:int]</td>
</tr>
<tr>
<td>Database Admin Report Data</td>
<td>[ProjectId:int</td>
</tr>
<tr>
<td>Security Admin Data</td>
<td>[AdminId:long int]</td>
</tr>
<tr>
<td>Security Operation Data</td>
<td>[AdminId:long int]</td>
</tr>
</tbody>
</table>

Where used/how used: Employee Delete Module( input )
                     Database Admin Operations( output )

Description:

Where used/how used: Employee Insert( input )
                     Employee Insert
                     Employee Delete( output )

Description:

Where used/how used: Backup Procedure( input )
                     Database Admin Operations ( output )

Description:

Where used/how used: Report Producer( input )
                     Database Admin Operations
                     Security Admin Operations( output )

Description:

Where used/how used: Employee Mail( input )
                     Report Producer ( output )

Description:

Where used/how used: Security Admin Module( input )
                     User Type Selector ( output )

Description:

Where used/how used: Security Admin Operations( input )
                     Security Admin Module( output )
Description:
[AdminId:long int|OperationRequest:int|SecurityAdminOperationsRight:bool]

Name: Assign Security Levels
alias: none
Where used/how used: Security Level Configuration( input )
                      Security Admin Operations( output )
Description:
[ProjectId:int|LevelId:int|IndexId:int|SecurityLevel:int]

Name: Security Admin Report Data
alias: none
Where used/how used: Employee Mail( input )
                      Report Producer( output )
Description:
[ProjectId:int|EmployeeId:long int|Text:string]

Name: Employee Operation Data
alias: none
Where used/how used: Employee Document Operations( input )
                      Employee Module( output )
Description:
[EmployeeId:long int|OperationRequestId:int]

Name: Approve Request
alias: none
Where used/how used: Approve Document( input )
                      Employee Document Operations( output )
Description:
[EmployeeId:long int|FileId:long int]

Name: Document Receive Request
alias: none
Where used/how used: Document Receiver( input )
                      Employee Document Operations( output )
Description:
[EmployeeId:long int|FileId:long int]

Name: Document Deliverance Request
alias: none
Where used/how used: Document Deliverer( input )
                      Employee Document Operations( output )
Description:
[EmployeeId:long int|FileId:long int]

Name: Employee Report Data
alias: none
Where used/how used: Project Reports( input )
                      Report Creator( output )
Description:
[EmployeeId:long int|ProjectId:int|FileId:long int]
Name : Approve Data
alias : none
Where used/how used : Employee( input )
                     : Approve Document( output )
Description :
[EmployeeId:long int|CustomerId:long int|FileId:long
int|ProjectId:int|ApprovalDate:date|ApprovalStatus:True]
           : ApprovalDate = [day|month|year]

Name : Received Document
alias : none
Where used/how used : PRequest
                  : PResponse( input )
                  : Document Receiver( output )
Description :
[PrequestId:long int]

Name : Delivered Document
alias : none
Where used/how used : PRequest
                  : PResponse( input )
                  : Document Receiver( output )
Description :
[PresponseId:long int]

Name : Project Leader Data
alias : none
Where used/how used : Project Leader Module( input )
                     : User Type Selector( output )
Description :
[ProjectLeaderId:long int]

Name : Project Leader Operations Data
alias : none
Where used/how used : Project Leader Operations( input )
                      : Project Leader Module( output )
Description :
[ProjectLeaderId:long int|OperationRequestId:int|ProjectLeaderOperationsRight:bool]

Name : Project Request
alias : none
Where used/how used : Project Leader Project Operations( input )
                      : Project Operations( output )
Description :
[ProjectLeaderId:int|RequestType:int|RequestTime:date]
           : RequestTime = [day|month|year]
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>alias</th>
<th>Where used/how used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation Request</td>
<td>[ProjectId:int</td>
<td>LevelId:int]</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Project Operations( output )</td>
</tr>
<tr>
<td>Update Request</td>
<td>[ProjectId:int</td>
<td>LevelId:int]</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Project Operations( output )</td>
</tr>
<tr>
<td>Creation Data</td>
<td>[ProjectId:int</td>
<td>LevelId:int]</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Project Update( output )</td>
</tr>
<tr>
<td>Update Data</td>
<td>[ProjectId:int</td>
<td>LevelId:int]</td>
<td>none</td>
</tr>
<tr>
<td>Document Request</td>
<td>[ProjectLeaderId:long int</td>
<td>RequestType:int]</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Project Leader Operations( output )</td>
</tr>
<tr>
<td>Create Request</td>
<td>[ProjectLeaderId:long int]</td>
<td>none</td>
<td>Document Creator( input )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Document Operations( output )</td>
</tr>
<tr>
<td>Document Data</td>
<td>[ProjectLeaderId:long int]</td>
<td>none</td>
<td></td>
</tr>
</tbody>
</table>
Where used/how used : Document( input )
                         Document Creator( output )

Description :
[DocumentType:string|TextLength:int|PictureSize:int*int|DocumentId:long int]

Name : PRequest Creation Request
alias : none
Where used/how used : PRequest Creator( input )
                        Document Operations( output )
Description : [ProjectLeaderId:long int]

Name : PResponse Creation Request
alias : none
Where used/how used : PResponse Creator( input )
                        Document Operations( output )
Description : [ProjectLeaderId:long int]

Name : Request Data
alias : none
Where used/how used : PRequest( input )
                        PRequest Creator( output )
Description :
[TextPlace:int*int|PicturePlace:int*int|RequestId:long int|DocumentIds:string]

Name : PResponse Data
alias : none
Where used/how used : PResponse( input )
                        PResponse Creator( output )
Description :
[TextPlace:int*int|PicturePlace:int*int|RequestId:long int|DocumentIds:string]

Name : Flowchart Request
alias : none
Where used/how used : Flowchart Operations( input )
                        Project Leader Operations( output )
Description : [ProjectLeaderId:long int]

Name : Flowchart Creation Request
alias : none
Where used/how used : Flowchart Creator( input )
                        Flowchart Operations( output )
Description : [ProjectLeaderId:long int]

Name : Flowchart Data
alias : none
Where used/how used : Flowchart( input )
                        Flowchart Updater
                        Flowchart Creator( output )
Description :
Name : Flowchart Update Request
alias : none
Where used/how used : Flowchart Updater( input )
Flowchart Operations( output )
Description : [ProjectLeaderId:long int]

Name : Employee Assignment Request
alias : none
Where used/how used : Assignment Operations( input )
Project Leader Operations( output )
Description : [ProjectLeaderId:long int]RequestType:int]

Name : Assignment Create Request
alias : none
Where used/how used : Employee Assignment( input )
Assignment Operations( output )
Description : [ProjectLeaderId:long int]

Name : Assignment Update Request
alias : none
Where used/how used : Employee Assignment Updater( input )
Assignment Operations( output )
Description : [ProjectLeaderId:long int]

Name : Assignment Data
alias : none
Where used/how used : Assigned Project( input )
Employee( output )
Description : [EmployeeId:long int]ProjectId:int]LevelId:int[IndexId:int]

Name : Employee Personal Data
Aliases : Employee Data
Where used/how used : To insert an employee to a project ( input )
Statistics and Emergency Situations ( output )
Description :
    [FldRef]Name|Surname|Address|StartDate|Telephone|Employee Status]
    Telephone = [areacode]local phone number]
    StartDate = [day]month|year]
    Address = [Street Name|Building Number|Door Number]

Name : Employee Login Data
Aliases : Account Data
Where used/how used : Login Screen ( input )
Description : [EmpId]UserName|Password]
Name : Employee Assignment Data
Aliases : Assignment Data
Where used/how used : Assigning Employees To Projects (input)
                      Statistics and Flowchart operations(output)
Description : [EmpId|ProjectId|LevelId|IndexId]
Name : Employee Reports
Aliases : none
Where used/how used : Reports and Statistics (input)
                      Document Approval and Rejection (output)
Description : [EmpId|ApprovalStatus|FileId|Date]
              Date = [day|month|year]

Name : Employee Statistics
Aliases : Statistical Data
Where used/how used : Reports and Statistics (input)
                      Document Approval and Rejection (output)
Description : [EmpId|WeekNumb|MonthNumb|YearNumb]

Name : Employee Mail Data
Aliases : Mail Data
Where used/how used : Employee Login Screen (input)
                      Employee Mail Operations (output)
Description : [MailId|EmpId|MailText|ReadStatus|SystemMailStatus|Date|From]
              Date = [day|month|year]

Name : Employee History Mail Data
Aliases : none
Where used/how used : Employee Mail Recovery (input)
                      Employee Mail Deletion (output)
Description : [MailId|EmpId|MailText|ReadStatus|SystemMailStatus|Date|From]
              Date = [day|month|year]

Name : Customer Personal Data
Aliases : Customer Personnel Data
Where used/how used : To insert an document to a flow (input)
Description : [FldRef|Name|Surname|Address|Telephone]
              Telephone = [areacode|local phone number]
              Address = [Street Name|Building Number|Door Number]

Name : Customer Login Data
Aliases : Customer Data
Where used/how used : Login Screen (input)
Description : [CustomerId|Username|Password]

Name : Current Request
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Statistics</td>
<td></td>
</tr>
<tr>
<td>Aliases</td>
<td>Statistical Data</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>System Defined Statistics (output)</td>
</tr>
<tr>
<td>ApprovalDate</td>
<td>= [day</td>
</tr>
<tr>
<td>Customer Mail</td>
<td></td>
</tr>
<tr>
<td>Aliases</td>
<td>Mail Data</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>Customer Personal Data (output)</td>
</tr>
<tr>
<td>Customer History Mail Data</td>
<td>= [day</td>
</tr>
<tr>
<td>User Defined Document</td>
<td>User Defined Flowchart</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>User Defined Prequest (input)</td>
</tr>
<tr>
<td>Description</td>
<td>[DocType</td>
</tr>
<tr>
<td>Pre-Defined Document</td>
<td>Pre-Defined Flowchart</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>Pre-Defined Prequest (input)</td>
</tr>
<tr>
<td>Description</td>
<td>[DocType</td>
</tr>
<tr>
<td>User Defined Prequest</td>
<td>Pre-Defined Project (input)</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>User Defined Project (input)</td>
</tr>
<tr>
<td>Description</td>
<td>[DocType</td>
</tr>
<tr>
<td>Description</td>
<td>User Defined Document (output)</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Name</td>
<td>Pre-Defined Request</td>
</tr>
<tr>
<td>Aliases</td>
<td>Pre-Request Data</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>Pre-Defined Project (input) Pre-Defined Document (output)</td>
</tr>
<tr>
<td>Description</td>
<td>[TextPlace</td>
</tr>
<tr>
<td>Name</td>
<td>User Defined Response</td>
</tr>
<tr>
<td>Aliases</td>
<td>Response Data</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>User Defined Project (input) User Defined Document (output)</td>
</tr>
<tr>
<td>Description</td>
<td>[TextPlace</td>
</tr>
<tr>
<td>Name</td>
<td>User Defined Flowchart</td>
</tr>
<tr>
<td>Aliases</td>
<td>Flowchart Data</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>Pre-Defined Project (input) Pre-Defined Document (output)</td>
</tr>
<tr>
<td>Description</td>
<td>[FlowId</td>
</tr>
<tr>
<td>Name</td>
<td>Pre-Defined Flowchart</td>
</tr>
<tr>
<td>Aliases</td>
<td>Flowchart Data</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>Pre-Defined Project (input) Pre-Defined Document (output)</td>
</tr>
<tr>
<td>Description</td>
<td>[FlowId</td>
</tr>
<tr>
<td>Name</td>
<td>User Defined Pdefinition</td>
</tr>
<tr>
<td>Aliases</td>
<td>Definition Data</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>User Defined Project (input)</td>
</tr>
<tr>
<td>Description</td>
<td>[Text</td>
</tr>
<tr>
<td>Name</td>
<td>Pre-Defined Pdefinition</td>
</tr>
<tr>
<td>Aliases</td>
<td>Definition Data</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>Pre-Defined Project (input)</td>
</tr>
<tr>
<td>Description</td>
<td>[Text</td>
</tr>
<tr>
<td>Name</td>
<td>User Defined Project</td>
</tr>
<tr>
<td>Aliases</td>
<td>Project Data</td>
</tr>
<tr>
<td>Where used/how used</td>
<td>Assigned Project Employee Assignment Data (input) User Defined PDefinition</td>
</tr>
</tbody>
</table>
User Defined Prequest
User Defined Flowchart
User Defined Report
System Defined Statistics
Document Information
User Defined Pdefinition ( output )

Description :
[LeaderId|ProjId|PreqIds|PresIds|SecMatrix|LvlMatrix|TimeMatrix]

Name : Pre-Defined Project
Aliases : Project Data
Where used/how used :
Assigned Project
Employee Assignment Data ( input )
Pre-Defined PDefinition
Pre-Defined Prequest
Pre-Defined Flowchart
Pre-Defined Report
System Defined Statistics
Document Information
Pre-Defined Pdefinition ( output )

Description :
[LeaderId|ProjId|PreqIds|PresIds|SecMatrix|LvlMatrix|TimeMatrix]

Name : Active Document
Aliases : Process Data
Where used/how used :
Employee Assignment Data ( input )
Current Request ( output )
Description :
[ProjId|FileId|LevelId|ApprovalId]

Name : Document Information
Aliases : Process Data
Where used/how used :
Customer Personnel Data
Current Request
User Defined Project
Pre-Defined Project ( input )
Description :
[EmpId|FileId|CustId|Date|ProjId]
Date = [day|month|year]

Name : System Defined Statistics
Aliases : none
Where used/how used :
User Defined Project
Pre-Defined Project ( input )
Customer Statistics
Employee Statistics ( output )
Description : [WeekNumb|MonthNumb|YearNumb|ProjId]

Name : User Defined Reports
Aliases : Report Data
Where used/how used : User Defined Project (input)
Description : [ProjId|ProCompDate|FileDialog|ProStartDate]

ProCompDate = [day|month|year]
ProStartDate = [day|month|year]

Name : System Defined Reports
Aliases : Report Data
Where used/how used :
Description : [ProjId|EmpId|Date|CustId|FileDialog|ApprovalStatus]

Date = [day|month|year]

Name : Project Assignment
Aliases : none
Where used/how used : Employee Personel Data (input)
User Defined Project (output)
Description : [ProjId|EmpId]

4.4 CONTROL FLOW DIAGRAM

4.4.1 INTERRUPTS:

Approval Interrupt:
This interrupt occurs when employee attempts to approve a document which he/she already approved. We will display an error message to the user in this case and then return back to approval state.

Login Interrupt:
This interrupt occurs when any user enters incorrect login. We will display an error message to the user in this case and then return back to login state.

Account Update Interrupt:
Employee account administrator causes this interrupt to occur when he/she tries to update account data with an empty request. In that case system will display an error message to her/him and then return back to the account update state.
**Account Create Interrupt:**
Employee account administrator causes this interrupt to occur when he/she tries to create account with a duplicate account data request. In that case system will display an error message to her/him and then return back to the account create state.

**Project Create Interrupt:**
Project Leader causes this interrupt to occur when he/she tries to create a project with an empty request. In that case system will display an error message to her/him and then return back to the project create state.

**Project Update Interrupt:**
Project Leader causes this interrupt to occur when he/she tries to create a project with a duplicate project data request. In that case system will display an error message to her/him and then return back to the project update state.

**PRequest Create Interrupt:**
Project Leader causes this interrupt to occur when he/she tries to create a prequest with an empty request. In that case system will display an error message to her/him and then return back to the prequest create state.

**PRequest Update Interrupt:**
Project Leader causes this interrupt to occur when he/she tries to create a prequest with a duplicate prequest data request. In that case system will display an error message to her/him and then return back to the prequest update state.

**PResponse Create Interrupt:**
Project Leader causes this interrupt to occur when he/she tries to create a presponse with an empty request. In that case system will display an error message to her/him and then return back to the presponse create state.

**PResponse Update Interrupt:**
Project Leader causes this interrupt to occur when he/she tries to create a response with a duplicate response data request. In that case system will display an error message to her/him and then return back to the response update state.

**Document Operations Interrupt:**
This interrupt occurs when an employee attempts to do any of the document operations with an absent document. We will display an error message to the employee in this case and then return back to document operations state.

**Statistics Report Interrupt:**
This interrupt occurs when an employee or project leader attempts to get any statistics with an empty statistics data. We will display an error message to the user in this case and then return back to statistics report state.

**Secure Mail Interrupt:**
This interrupt occurs when any kind of user (also web users) attempts to send an empty mail. We will display an error message to the user in this case and then return back to secure mail state.

*(For Control Flow Diagram See Appendix)*

**4.5 STATE TRANSITION DIAGRAM**

In State Transition Diagram, we showed that if an interrupt occurs the states change into rejection state to display error message and there exists return to every state from rejection state. Also from the secure mail state, there exists return to employee, project leader and administrator states.

**Approval:**
If any of the employees attempt to approve any document, this means he/she is in approval state. If he/she tries to approve an approved document approval is rejected, if approval is applicable he/she returns to the employee state.
Statistic Reports:
When a user or system automatically requests to create report statistics, system gets into the Statistics Report State. If data for statistics is empty request is rejected, also if employee or project leader data for creating statistics is ok, then statistics report is created. Finally system returns to employee or project leader state according to the user type.

Employee:
An employee can create a mail and make a request to send this mail via secure mail. If the mail created is empty, request is rejected (system gets into rejection state). Whenever mail is applicable to send, system gets into secure mail state to send the mail.

Secure Mail:
System gets into this state when any users of the system attempts to send an applicable mail. Then the mail is send successfully in this state. Finally system returns to the users state according to the user type.

Login:
Firstly, system is in this state since none of the users can do any work before login. If entered login data id incorrect system continues to stay in this state. If login data is correct, system can change state to project leader, employee or administrator by looking which of these three has login.

Administrator:
This state is invoked when any of the three admins login. Account update request can be done in this state by employee account admin which causes system to get into account update state. Moreover, system gets into account create state when account create request is made.
**Account Create:**

System gets into this state when an account create request is done. If empty account data exists system invokes rejection, whenever account creation is done successfully system automatically gets back into administrator state.

**Account Update:**

System gets into this state when an account update request is done. If duplicate account data exists for update, system invokes rejection, whenever account update is done successfully system automatically gets back into administrator state.

**Document Operations:**

This state is invoked when an employee attempts to make document operations. If the document is absent system rejects to do operation and gets into rejection state but if the document is applicable system successfully gets into employee state to give the chance to employee make the operation.

**Project Operations:**

System jumps to this state when project leader makes a request to do project operations. If duplicate of project condition occurs, system rejects the request, whenever project is applicable system successfully gets into project leader state to give the chance to project leader make the operation.

**Project Leader:**

This state is invoked when a project leader login. A project leader can make project create, project update, flowchart create, flowchart update, presponse create, presponse update, prequest create, prequest update requests. Each request causes system to get into its corresponding state which are same named in the state transition diagram.
**Project Create:**
System gets into this state when project leader makes request to create project. If duplicate of project condition exists system invokes rejection, whenever project creation successfully occurs system turns back to project leader state.

**Project Update:**
System gets into this state when project leader makes request to update project. If empty project condition exists system invokes rejection, whenever project update successfully occurs system turns back to project leader state.

**Presponse Create:**
System gets into this state when project leader makes request to create presponse. If duplicate of presponse condition exists system invokes rejection, whenever presponse creation successfully occurs system turns back to project leader state.

**Presponse Update:**
System gets into this state when project leader makes request to update presponse. If empty presponse condition exists system invokes rejection, whenever presponse update successfully occurs system turns back to project leader state.

**Prequest Create:**
System gets into this state when project leader makes request to create prequest. If duplicate of prequest condition exists system invokes rejection, whenever prequest creation successfully occurs system turns back to project leader state.

**Prequest Update:**
System gets into this state when project leader makes request to update prequest. If empty pres prequest condition exists system invokes rejection, whenever prequest update successfully occurs system turns back to project leader state.
(For State Transition Diagram See Appendix)
5 OBJECT ORIENTED ANALYSIS

5.1 CLASS DIAGRAMS

In the class diagram, which is used for static design view of a system, we showed the set of classes, collaborations, interfaces and their relationships. We have thirty-one classes which are Employee Personal Data, Assigned Project, Employee Login Data, Employee Mail Data, Employee History Mail Data, Active Document, Employee Assignment Data, Predefined PDefinition, User Defined PDefinition, User Defined Project, Predefined Project, Current Request, PreDefined PRequest, PreDefined PResponse, PreDefined Flowchart, PreDefined Document, User Defined PRequest, User Defined PResponse, User Defined Flowchart, User Defined Report, System Defined Statistics, System Defined Reports, Employee Reports, Document Information, User Defined Document, Employee Statistics, Customer Statistics, Customer Personal Data, Customer Mail, Customer Login Data, Customer History Mail Data.

At the top of the diagram there exists the Employee Personal Data which is connected to other five classes with an aggregation meaning that five classes construct our Employee Personal Data. One of the thes five classes called Assigned Project is connected to another class User Defined PDefinition with an association indicating that User Defined Project stores the information about Assigned Project. Moreover they are connected with 1-* relationship. One more association exists between Active Document and Employee Assignment Data indicating that active document is approved by the employees stored at the Employee Assignment Data. Associations between the Employee Assignment Data and User Defined Project and between Employee Assignment Data and Predefined Project are n to n relations meaning that Employee Assignment Data can contain more than one User Defined Projects and Predefined Projects and vice versa. User Defined Project is connected to eight classes with an aggregation which are User Defined PRequest, User Defined PResponse, User Defined Flowchart, User Defined Report, System Defined Statistics, System Defined Reports, Employee Reports, Document Information.
Another aggregation exists between the Customer Personal Data and Customer Mail, Customer Login Data, Customer History Mail Data. Customer Personal Data is the whole part and others are the components of it.


(For Class Diagram See Appendix)

5.2 COLLABORATION DIAGRAM:

In the first collaboration diagram, you see that when Customer is login then customer object is created and after this with setGUI ( ) GUI is prepared for customer. When customer is login the processes he/she can do are showed which are send mail, read mail, delete mail and current request. These processes are related to their depending database tables. (At delete process since we store the mails which are deleted by customer to Customer History Mail Data, delete process causes another process shown by 2.3.1 stare mail)

In the second collaboration diagram, you see that when Manager is login then manager object is created and after this with setGUI ( ) GUI is prepared for manager. When customer is login the processes he/she can do are showed which are send mail, read mail, delete mail, report display, project leader assignment and project assignment. These processes are related to their depending database tables. (Again we store the deleted mails so explanations done above for customer are also true for Manager)

In the third collaboration diagram, you see that when Security Admin is login then sa object is created and after this with setGUI ( ) GUI is prepared for security admin. When
security admin is login the processes he/she can do are showed which are send mail, read mail, delete mail and report create and security level configuration. These processes are related to their depending database tables. (Again we store the deleted mails so explanations done above for customer are also true for Security Admin)

In the fourth collaboration diagram, you see that when Database Admin is login then da object is created and after this with setGUI ( ) GUI is prepared for database admin. When database admin is login the processes he/she can do are showed which are send mail, read mail, delete mail and report create and employee insert, employee delete. These processes are related to their depending database tables. (Again we store the deleted mails so explanations done above for customer are also true for Security Admin)

In the fifth collaboration diagram, you see that when Employee Account Admin is login then eaa object is created and after this with setGUI ( ) GUI is prepared for employee account admin. When employee account admin is login the processes he/she can do are showed which are send mail, read mail, delete mail and report create and employee account create and employee account update. These processes are related to their depending database tables. (Again we store the deleted mails so explanations done above for customer are also true for Security Admin. Also we store the changes done in employee accounts. These process is shown by 2.4.1 and 2.5.1)

In the sixth collaboration diagram, you see that when Employee is login then e object is created and after this with setGUI ( ) GUI is prepared for employee. When employee is login the processes he/she can do are showed which are send mail, read mail, delete mail and report create, document approval, document receive and document delivery. These processes are related to their depending database tables. (Again we store the deleted mails so explanations done above for customer are also true for Employee. Also we store the documents delivered and received by employee. These processes are shown by 2.6.1 and 2.7.1)

In the seventh collaboration diagram, you see that when Project Leader is login then pl object is created and after this with setGUI ( ) GUI is prepared for project leader. When project leader is login the processes he/she can do are showed which are send mail, read
mail, delete mail, employee assignment, employee assignment update, project create, project update, document create, prequest create, presponse create, flowchart create and flowchart update. These processes are related to their depending database tables. (Again we store the deleted mails so explanations done above for customer are also true for Employee. Also we store the data of employee assignment, employee assignment update, project create, project update. These processes are shown by 2.4.1, 2.5.1, 2.6.1, 2.7.1, 2.6.2, 2.7.2, 2.6.3 and 2.7.3 )

(For Collaboration Diagram See Appendix)

5.3 SEQUENCE DIAGRAMS

The first sequence diagram is for customer. It includes the customer object, GUI created for customer, and customer related databases (Customer Mail, Customer History Mail Data, Document Information). When customer logs in customer object is created and GUI is created for this customer object. Then you see the processes which the customer can do from GUI which are send mail, read mail, delete mail and current request. These processes are connected to their related database tables (where info is taken from to do the process) with arrows. During delete mail process another action is activated since our system stores the mails deleted by customer. Finally, when customer is logout the customer object is destroyed.

The second sequence diagram is for security admin. It includes the security admin object, GUI created for security admin, and security admin related databases (Employee Mail, Employee History Mail Data, Project). When security admin logs in security admin object is created and GUI is created for this security admin object. Then you see the processes security admin can do from GUI which are send mail, read mail, delete mail and report create, security level configuration. These processes are connected to their related database tables (where info is taken from to do the process or info obtained is stored) with arrows. During delete mail process another action is activated since our system stores the mails deleted by security admin. Finally, when security admin is logout the security admin object is destroyed.
The third sequence diagram is for database admin. It includes the database admin object, GUI created for database admin, and database admin related databases (Employee Mail, Employee History Mail Data and Employee). When database admin logs in database admin object is created and GUI is created for this database admin object. Then you see the processes database admin can do from GUI which are send mail, read mail, delete mail and report create, employee insert, employee delete. These processes are connected to their related database tables (where info is taken from to do the process or info obtained is stored) with arrows. During delete mail process another action is activated since our system stores the mails deleted by database admin. Finally, when database admin is logout the database admin object is destroyed.

The fourth sequence diagram is for employee account admin. It includes the employee account admin object, GUI created for employee account admin, and employee account admin related databases (Employee Mail, Employee History Mail Data and Employee Login Data, Employee Personal Data). When employee account admin logs in employee account admin object is created and GUI is created for this employee account admin object. Then you see the processes employee account admin can do from GUI which are send mail, read mail, delete mail and employee account create, employee account update. These processes are connected to their related database tables (where info is taken from to do the process or info obtained is stored) with arrows. During delete mail process another action is activated since our system stores the mails deleted by employee. Also during the employee account create; employee account update processes store data activations are created to store the data of employee account admin. Finally, when employee account admin is logout the employee account admin object is destroyed.

The fifth sequence diagram is for employee. It includes the employee object, GUI created for employee, and employee related databases (Employee Mail, Employee History Mail Data and Employee Statistics, Project Report, Prequest, Preponse). When employee logs in employee object is created and GUI is created for this employee object. Then you see the processes employee can do from GUI which are send mail, read mail,
delete mail and report create, document delivery, document approval, document receive. These processes are connected to their related database tables (where info is taken from to do the process or info obtained is stored) with arrows. During delete mail process another action is activated since our system stores the mails deleted by employee. Finally, when employee is logout the employee object is destroyed.

The sixth sequence diagram is for manager. It includes the manager object, GUI created for manager, and manager related databases (Employee Mail, Employee History Mail Data and Project Report, Assigned Project, Employee Assignment). When manager logins manager object is created and GUI is created for this manager object. Then you see the processes manager can do from GUI which are send mail, read mail, delete mail and report display, project leader assignment, project assignment display. These processes are connected to their related database tables (where info is taken from to do the process or info obtained is stored) with arrows. During delete mail process another action is activated since our system stores the mails deleted by manager. Finally, when manager is logout the manager object is destroyed.

Finally, the sixth sequence diagram is for project leader. It includes the project leader object, GUI created for project leader and project leader related databases (Employee Mail, Employee History Mail Data and Project, Prequest, Presponse, Assigned Project, Employee Assignment Data, Document, Flowchart). When project leader logins project leader object is created and GUI is created for this project leader object. Then you see the processes project leader can do from GUI which are send mail, read mail, delete mail, employee assignment, employee assignment update, project create, project update, document create, prequest create, presponse create, flowchart create, flowchart update. These processes are connected to their related database tables (where info is taken from to do the process or info obtained is stored) with arrows. During delete mail process another action is activated since our system stores the mails deleted by project leader. Also during the employee assignment, employee assignment update, project create, project update processes store data activations are created to store the data of this processes. Finally, when project leader is logout, the project leader object is destroyed.

(For Sequence Diagram See Appendix)
6. ARCHITECTURAL DESIGN

Software Design is a multi-step process in which representations of data, program structure, and interface characteristics and procedural detail are formed from information requirements. Therefore, we can argue that design process is information driven. Since, the data is gathered in the requirements analysis step, we will be able to derive “Architectural Design”.

Architectural design represents the structure of data and program components that are required to build a computer-based system. In that step the components that constitute the system are defined. Indeed, the structure and properties of all components are decided. The interrelationships that occur among all the architectural components are also defined.

The architectural design step begins with data design. Data design should create a model of data that is represented at a high level of abstraction. Since our investigations show that the architecture of data has an influence on the architecture of the software that will process it, the data design step is completed with attentive labor. All the data objects that are defined during the requirement analysis step are modeled using entity relationship diagrams and a detailed data dictionary.

Since the data design step is completed with great care, the architectural style of the software could be explained. There exist many architectural styles which describes a system category that encircle a set of components, a set of connectors that enable coordination among the components, constraints that define how components can be integrated. Since all the patterns have their own properties, we have to select the model that is most appropriate to our software. The Data Flow Management System that we will implement requires many users that will connect to the system using the web or network. All the data related to those users will be stored in some databases and once a user connects the system some data will be changed. What is the most important aspect of the system is that, the system will run in a way that is defined before. That means all the documents will run in a sequence from one component to another. That is actually, the Data-flow architecture in which the input data is transformed through a series of components to the output. Our pattern will be a pipe and filter pattern. Each filter will
work independently of those other filters and expects a data input of a certain form and produces data output of a specified form.

6.1 TRANSFORM MAPPING

In the Data Flow Management System we are going to implement, information & data will be taken as an external data. That data input needs to be converted into an internal data to be processes through the software components. Finally, the data is to be converted into an external data to be shown as data output. Since the explained methodology above is of transform flow, we will use transform mapping when we will map the requirements into the software architecture.

1. Review the fundamental system model

When the Data Flow Diagrams are investigated, it is clearly seen that the system specifications and software requirements specifications are both satisfied.

2. Review and refine data flow diagrams for the software

Level 3 Data Flow Diagram is derived from Level 2 Data Flow diagram to improve the detail. At level 3 DFD each transform exhibits relatively high cohesion. Transactions have processes that perform single and distinct functions that can be implemented as modules. Looking at the level 3 Data Flow Diagram and the requirements of the system the modules in our system are defined. In the system we are going to implement, there are 11 modules. These are Web-based user login module, Network user login module, Employee module, Manager Module, Project Leader module, Security Administrator module, Employee Account Administrator, Database Administrator, Secure Mail module, Reports module and Statistics modules. The general structure of the modules is explained below:

Web-based user login module

This module will be designed to provide the “customers” a secure login process. In this module there are 6 sub modules and a display. These are Web Login Menu, Registered User Module, Request Generator, System Operations, Quick Response
Module, Process Starter, and Login Report Display. In this module there will be Web user Login Menu that will take the User Data as input from the user and generates Authorization Data that will pass through Registered User Module. That module will generate User Request that will pass through Request Generator. User Request is used to represent the user request which is mail request, quick response or document insertion into one of the projects. Request Generator generates the Generated request that passes to the System Operations. In that sub module 3 internal data are produced. Document that passes through Process Starter, Information Request that passes through Quick Response Module and Mail Request that passes through Secure Mail module which is one of the eleven modules of the software system. The Quick Response module will generate the Report Data to be displayed at the Report Display. The external input data of the module is simply the User Data, which is provided by the user. The internal data are Customer Data, Authorization Data, Customer Personal Data, Definition Data, User Request, Generated Request, Mail Request, Document, Information Request, Process Data, Statistical Data, and Report Data. The external data that are sent as data output are the User Login Status and the Report Data.

Network user login module

Similar to the web based login module, that module has also the same goal, which is to provide a secure login process, however this module’s target is not the web based user but the network user. In this module there are 2 sub modules and two displays. These are Network Login Menu, User Type Selector, Login Status Display, and Selection Status Display. In this module Employee User Data is provided by all types of employees in the system to connect the system. That data is passed through the Network Login Menu and Authorization Data is generated. With the Employee Data and Assignment Data that are taken from the databases User Type Selector generates the data of each employee type which actually are; Manager Data, Employee Data, Project Leader Data, Security Administrator Data, Employee Account Administrator Data, and Database Administrator Data. The external input data of the module is simply the Employee User Data, which is provided by the user. The internal data are Account Data, Authorization Data, Employee Personal Data, Assignment Data, Manager Data, Employee Data, Project Leader Data, Security Administrator Data, Employee Account Administrator Data, and Database
Administrator Data. The external data that are sent as data output are Selection Status Data and Login Status Data.

Employee module

The Employee module will be designed to provide the Employee User the general operations he is going to execute. In this module there are 10 sub modules and 4 displays. The Employee Data is sent into the Employee Module as input data and it is converted into Employee Operation Data. That data is passed through Employee Document Operations. That sub module creates four Data, namely Report Request, Approve Request, Document Receive Request and Document Deliverance Request. Those data are sent to Report Creator, Approve Document, Document Receiver and Document Deliverer respectively. Report Creator creates the Report Display Data and sent to the Report Creation Display Creator. Approve Document sub module creates Approval Display Data and sent to Approval Display Creator. Document Deliverer creates Document Receive Display Data and sent to Document Receive Display Creator. Document Deliverer sub module generates the Document Delivery Display Data and that data is passed through Document Delivery Display Creator. The status data are created to provide the external entities that are screens in our case the necessary data. Those necessary data are Report Creation Display Status, Approval Display Status, Document Receive Status, and Document Delivery Status. These data are generated by the sub modules Report Creation Display Creator, Approval Display Creator, Document Receive Display Creator, Document Delivery Display Creator respectively. The displays provide the employee information after the process ends. The displays are Report Creation Display, Approval Display, Document Receive Display, and Document Receive Display. The internal data that processed inside that module are Employee Operation Data, Report Request, Approve Request, Document Receive Request, Document Deliverance Request, Report Display Data, Approval Display Data, Document Receive Display Data, Received Document, Document Delivery Display Data, Report Data, Approve Data, Received Document, Received Document, Delivered Document. The external data are Report Creation Display Status, Approval Display Status, Document Receive Status, and Document Delivery Status.
Manager module

The Manager Type user is one of the users in the system that we are going to implement. The operations that are executed by that user are handled with that module. After the manager module data is provided to the module, the module creates three types of data that are sent to sub modules. Those data are Report Request, Project Data, and P.L. Assignment. The Report Request Data is sent to Report Creator sub module. The Project Data is sent to Project Operations sub module and P.L. Assignment is sent to P.L. Assignment sub module. Those modules create some data that are sent to displays or other sub module that generates the display. Report Creation Data is created by the Report Creator and sent to Report Creation Display. Project Assignment Display Request is created by Project Operations and sent to Project Assignment Show sub module that provides necessary data Assignment Display Data to Project Assignment Display. PL Assignment Data is created by P.L. Assignment and sent to PL Assignment Display Controller sub module which provides necessary data PL Display Data to PL Assignment Display. The internal data of the module are Report Request, Project Data, P.L. Assignment, Report Creation Data, Project Assignment Display Request, and PL Assignment Data. The external data are Assignment Display Data, P.L. Display Data. In that module there are three display with the names: Report Creation Display, Project Assignment Display, PL Assignment Display.

Project Leader module

Project Leader is another user type of the system. The Project Leader Module represents that user and his operations. Once the Project Leader Data is received the Operation Data is created. That data is sent Project Leader Operations sub module. That sub module produces four types of data namely; Project Request, Document Request, Flowchart Request, Assignment Request. Project Request is sent to Project Operations to handle the project operations. Document Request, Flowchart Request is sent Document Operations to handle the document operations. Flowchart Request is sent to Flowchart Operations to handle the flowchart operations. Finally, Assignment Request is sent to Assignment Operations. Project Operations sub module creates Creation Request, Update Request. Creation Request is sent to Project Creator sub module, Update Request
is sent to Project Updater. Since there are three types of document operations which are document creation, prequest creation or presponse creation, there are three data to represent them with the names Create Request, Prequest Creation Request, Presponse Creation Request. Create Request is sent to Document Creator, Prequest Creation Request is sent to Prequest Creator and Presponse Creation Request is sent to Presponse Creator. The Flowcharts Operations sub module creates two data which are Flowchart Creation Request and Flowchart Update Request. These data are sent to Flowchart Creator and Flowchart Updater sub modules respectively. The Assignment Operations sub module creates Assignment Request and Assignment Change Request. The Assignment Request is sent to Employee Assignment sub module and the Assignment Change Request is sent to Employee Assignment Updater sub module. There is a message display which is used to show the user the final status and information on the operation he executes. This Message Display needs Message Display Data created by the Message Creator sub module. The Message Creator sub module needs Message Data that is passed from Project Operations, Document Operations sub module, Flowchart Operations sub module or Assignment Operations sub module. The internal data in the module are Operation Data, Assignment Request, Flowchart Request, Document Request, Project Request, Assignment Change Request, Assignment Request, Assignment Data, Assignment Request, Flowchart Update Request, Flowchart Creation Request, Flowchart Data, Presponse Creation Request, Prequest Creation Request, Create Request, Document Data, Prequest Data, Presponse Data, Creation Request, Update Request, Project Data Message Data. The external data are Message Display Data, Mail Request.

**Security Administrator module**

Security Administrator is one of the three administrators in the system. The Security Administrator Module is designed to represent that employee type. There are 4 sub modules and a display in that module. These are Security Admin Module, Security Administrator Operations, Security Level Configuration, Report Producer, and Message Display. The Security Admin Module takes the Security Admin Data and creates Operation Data in which the operational data of the administrator is stored. This data is directly sent to Security Administrator Operations and that sub module creates Assign
Security Levels, Report Send Request Data with respect to the request of the security administrator. Assign Security Levels Data is sent to Security Level Configuration to assign security levels in the projects. Report Send Request Data is sent to Report Producer sub module in which the reports are produced. Report Producer and Security Level Configuration updates the related databases and creates Message Data that is used to create the Message Display of the operation after the execution. Finally, the internal data are Operation Data, Assign Security Levels, Report Send Request Data, Report Data, Security Data. The external data are Security Admin Data, Message Data, Mail Request.

Employee Account Administrator

Employee Account Administrator is one of the three administrators who is responsible of account operations of the employees. In that module there are 4 sub modules and a display. These are Employee Account Admin Module, Account Creator, Account Update, Operation Message Creator, Message Display. Employee Account Admin Data is passed through the Employee Account Admin Module and New Account Request and Update Request are created. As the names refer the first data is the representation of new account creation operation of the administrator and the latter data will be used to represent the modifications in the employees` account. New Account Request is sent to the Account Creator sub module and the Update Request is sent to the Account Update sub module. Account Creator creates the Account Data and the Account Updater creates the New Account Data and both of the sub modules make the necessary modifications in the related databases. Operation Message Creator sub module takes Message Data from Account Creator and Account Updater sub modules. The external data are; Employee Account Admin Data, Mail Request and Message Data. The internal data of that module are; Update Request, New Account Request, Account Data, New Account Data.

Database Administrator

Database Administrator will be responsible of the database operations like back up. The module named as Database Administrator module will be the representation of that user type. In that module there are 6 sub modules and a display. These are Database
Administrator Module, Database Administrator Operations, Employee Delete, Employee Insert, Backup Procedure, Report Producer, Message Display. Security Admin data is sent to the Database Administrator Module and Operation Data is created in which the operational data of the database administrator is stored. That data is sent to Database Administrator Operations in which the operation selection is made. Four types of data are created here with respect to the sub module it will be sent to. Delete Employee Request, Insert Employee Request, Back up Request, Report Send Request are created. Delete Employee Request is sent to Employee Delete sub module. Insert Employee Request is sent to Employee Insert sub module, Backup Request is sent to Backup Procedure sub module. Report Send Request is sent to Report Producer sub module. The sub modules modify the relevant databases. The internal data in the module of Operation Data, Delete Employee Request, Insert Employee Request, Backup Request, Report Send Request, Employee Data. The external data are Message Data and Report Data.

Secure Mail module

As described in the Requirements Analysis Report, there will be a secure mail module which will be used to provide a mailing process among the users and the customers in a secure way. The Secure Mail Module is the module which will be designed to achieve that goal. In the Secure Mail Module, there are 5 sub modules and a display. These are Secure Mail, Mail Delete, Mail Operations, Mail Read Module, Mail Composer, Mail Status Display. After the Mail Request is passed into the module, Secure Mail Request is generated. This data is sent to Mail Operations sub module. Mail Operations sub module creates three data which are delete request, compose request, read request. These are sent to Mail Delete, Mail Composer, and Mail Read Modules respectively. Those modules will modify the relevant databases and will create data with the name Status Data which will be used to create the Message Display after the requests end.
Reports module

Reports are very important parts of the software that we are going to implement. The Report Module will be implemented to handle the report actions. In this module there are three sub modules and a display. Report Operations, Report Creator, Report Evaluator, Report Display are the sub modules and the display of the Reports Module. Once the Report Request Data enters the Report Operations, Report Creation Request and Report Evaluation Request are generated. These two data are directly sent to Report Creator sub module and Report Evaluator sub module. Report Evaluator creates the Report Data which is needed to create the Report Display. The external data are Report Request and the Report Data. The internal data are the Report Creation Request, Report Evaluation Request, Project Report Data

Statistics modules

Statistics are another important concept of our software, in which some statistics that are explained in the requirements analysis report are stored. The Statistics Module will be implemented to handle the statistics operations, which are executed by the system itself. In the Statistics Module there are 2 sub modules and a display. These are Statistics Operations, Statistic Generator and Statistics Display. Once the Statistics Request comes to the Statistics Operations, Statistics Operation is created. Statistics Operation Data is directly passed into the Statistic Generator. The Statistic Generator creates Statistics Display Request which is needed to create the Statistics Display. The internal data in the module are Statistics Operation, Statistical Data, Statistics Display Request. The external data are statistics request.

(For Modules See Appendix)

3. Determine whether the DFD has transform or transaction flow characteristics
As explained before our system has transform flow characteristics therefore, we will use transform mapping.

4. **Isolate the transform center by specifying incoming and outgoing boundaries**
   For Boundary Diagram, look Appendix.

5. **Perform First Level factoring**
   For First Level Factoring Diagrams, look Appendix

6. **Perform Second Level factoring**
   For Second Level Factoring Diagrams, look Appendix

7. **Refine the first iteration architecture using design heuristics for improved software quality**
   For First Iteration Program Structure Diagrams look Appendix
7. INTERFACE DESIGN
The interface allows the user interact with the network or the web. The first and foremost purpose of the interface is to translate information between the document flow management system and the user. Without the interface, customers or the employees will not easily see the flow of the documents.

The interface allows the user:

- fast and easy interface between the users of system
- understandable menus and languages on the user’s own accounts
- learn the place of the document he needs
- store mails in an organized way
- configure security levels easily (for administrators)
- create reports and statistics
- all types of data modification from the database according to the authorization
- document delivery, approval and receive

7.1 INTERFACE DESIGN RULES
Interface design focuses on three areas of concern:

- The design of interfaces between software modules;
- The design of interfaces between the software and other nonhuman producers and consumers of information (i.e., other external entities);
- The design of the interface between the user and the computer

Human-machine interface specification will be done by two servers: mail server and application server. Our mail server will encrypt, decrypt and store mails in the database via secure mail to have reusable and secure mailing system for the users. Since our system has many processes to run and many databases to handle, we are going to use application server to manage and schedule the processes. The system has many applications to execute and these applications are the backbone of our system. The data and the processes are used by these applications. Since our system has many processes to run and many databases to handle, we are going to use this server to manage and schedule the processes. For example, if a document flow is stuck at some employee the applications on this server detects the employee and sends reports to the related
employees by using the mail server and it stores these reports in the specific database which is implemented in the file and database server.

Internal interface between the user and the computer will be ensured by two servers: web server and the file&database server. The web user or the network user will ensure his account by passing a login procedure and then see his own account. Our web server will enable user interface to the web-based users due to their profile settings. Since our system has many databases, we are going to use file&database server to compose an interface between the processes created by the system. This server must be more robust with respect to the other servers.
7.2 EXAMPLES FOR GUI
An example menu for the customer and the employee login screen is prepared using Delphi.
Two examples for the personal pages are shown below for different types of employees. Both are prepared in Delphi.