Software Design Document ————

Ahmed Hani,Hend Mohamed,Rowan Omar,Hossam Badr

May 15, 2017

1 Introduction

1.1 Purpose

This Software Design Document provides the design details of Al shams Company for Housing and Construction System.

1.2 Scope

This document contains a compelete description of the design of the system.This software system will be a Web Application for a housing and construction company. This system will be designed to maximize the companys productivity by providing tools to assist employees to manage financial and management issues of the company and allow customers to contact the company easily. By maximizing the employees work efficiency and production the system will meet the employees needs while remaining easy to understand and use. More specifically, this system is designed to allow accounting stuff to manage accounting issues such as contracts, transactions, payments, cheques, account statements and letters of guarantee. An administrator also uses the system in order to administer the system and keep the information accurate. It allows the technical stuff to manage time plans, projects plans, cost estimates and demodulators. The software will facilitate communication between customers and the company so, the customer can register for an account and apply for a new project. The system has a user friendly and usable interface, supported by a well designed relational database.

1.3 Definitions, Acronyms, Abbreviations

Term	Definition
Software Design Document (SDD)	Used as the primary medium for communicating software design information.
MVC	Model view Controller

1.4 Overview of Document

This document wil have all of design and architecture for the system.then we have the Class Diagram,Sequence Diagrams,Data flow diagram , Database Design and we have UI section.

2 System Overview and Architecture

2.1 Overview

The system used in MVC architecture and using

2.2 Architecture Design

Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Dont go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram as well.

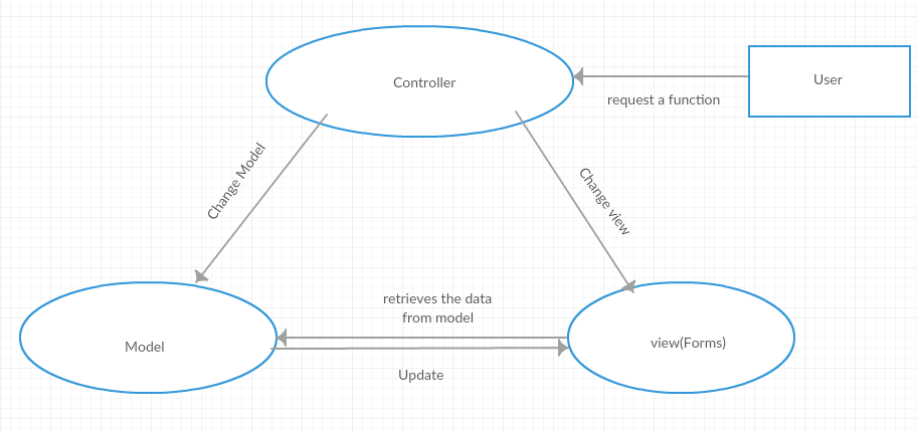

Figure 1: Architectural Design

3 Decomposition Description

In this section for a functional description, put top-level data flow diagram (DFD). For an OO description, put object diagrams and sequence diagrams.

Supplement with text as needed.

3.1 Class Diagram

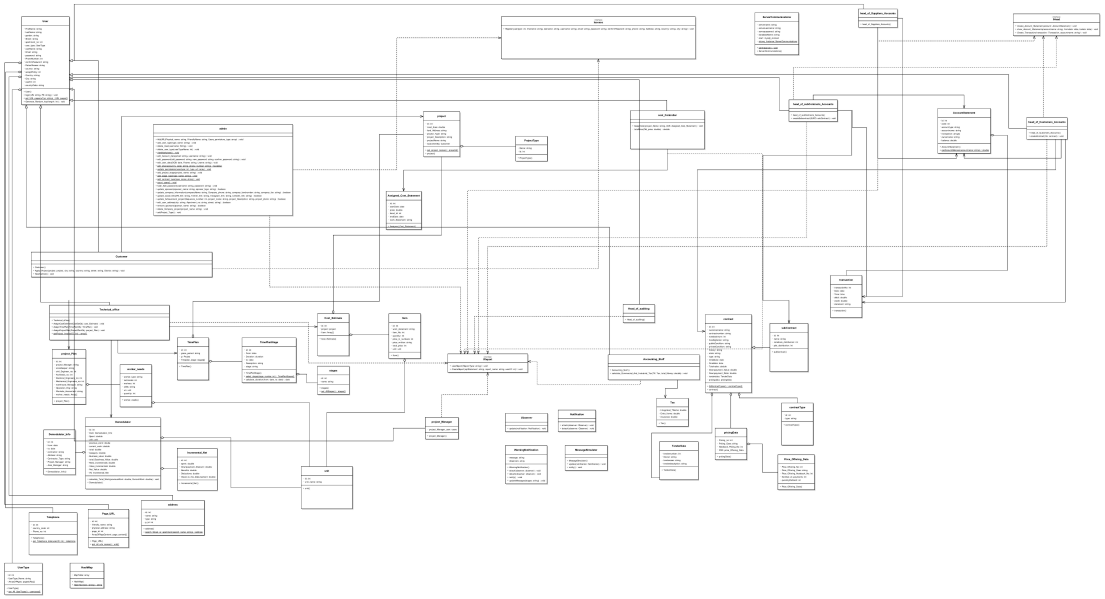

Figure 2: Class Diagram

3.2 Sequence Diagrams

3.3 Data flow diagram

3.4 Database Design

The database has EAV design and many Dynamic design.

4 Human Interface Design

4.1 Overview of User Interface

if the user has access as admin. he will be log in and can add user,projects,permissions and user types.he also can show reports and Mark sheets. he can add new forms.

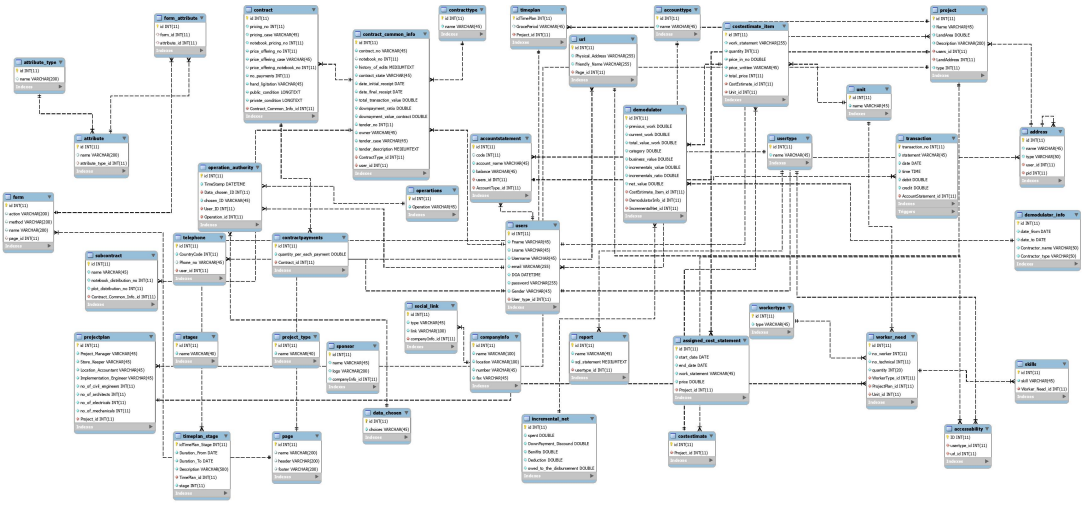

Figure 3: Class Diagram

4.2 Screen Images

5 Requirements Traceability

Provide a cross reference that traces components and data structures to the requirements. Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

6 Test case Details

6.1 Apply For new Project

Happy Scenario:
Input: Project Info (Land Area, Land Address, Project Type, Project Description, Customer Name, Address, Phone Number).
Output: Project added successfully.
Bad Scenario:
Input: Customer entered a name of an existing project.
Output: Alert Project already exists.

6.2 Add Cost Estimate

Happy Scenario:
Input: Choose Project Name, type and add Cost Estimate(Items Unit, Item

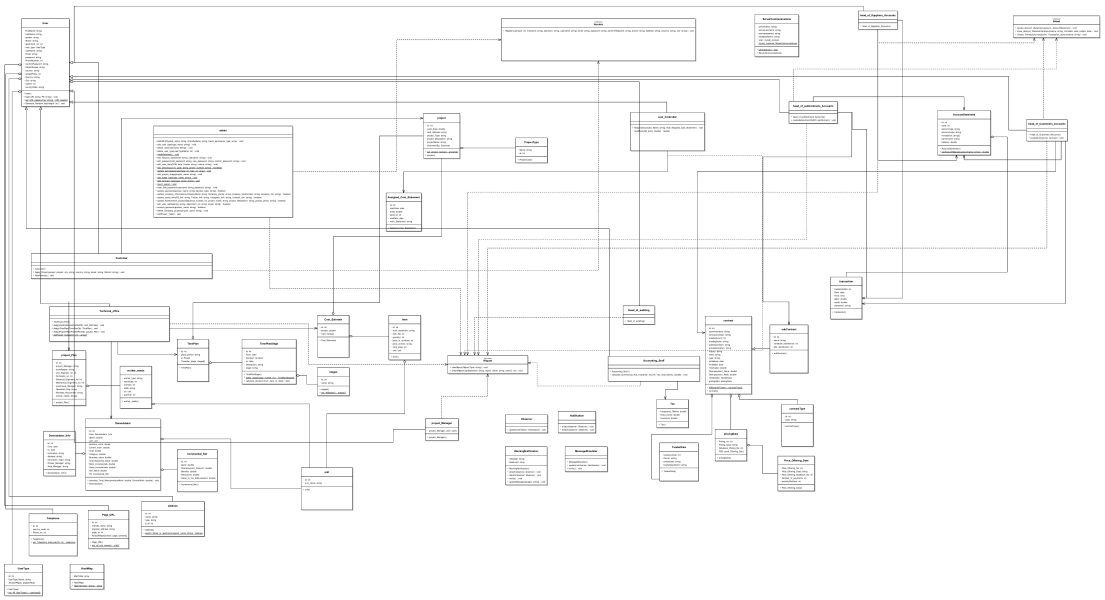

Figure 4: Class Diagram

Number, Material and Work Release, Unit Price In Number, Unit Price Written, Total Price).
Output: Cost Estimate added successfully.
Bad Scenario:
Input: Add an existing cost estimate to the same project.
Output: Alert Cost Estimate already exists.

6.3 Add Time Plan Test Case 1

Happy Scenario:
Input: Choose Project Name, type, owner and Time Plan Stages (Five Stages, Stage Description, and Stage Duration (from/to)).
Output: Time Plan added successfully.
Bad Scenario:
Input: Add an existing Time plan to the same project.
Output: Alert Time Plan already exists.

6.4 Add Time Plan Test Case 2

Bad Scenario:
Input: Adding a stage with a deadline that has been expired.

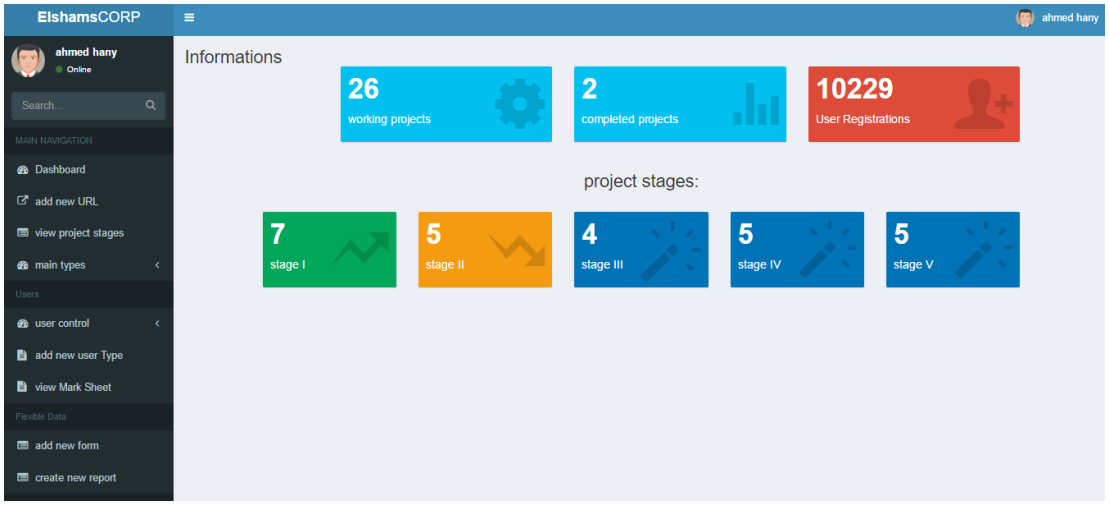

Figure 5: Admin Page

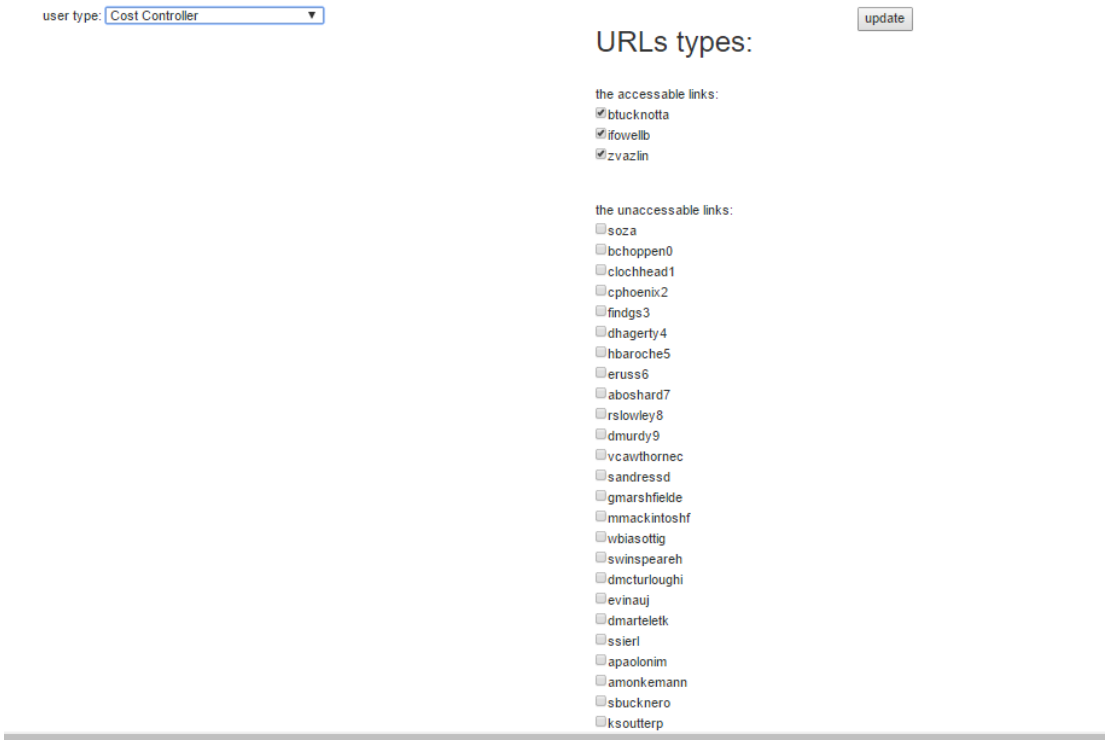

Figure 6: permissions Page

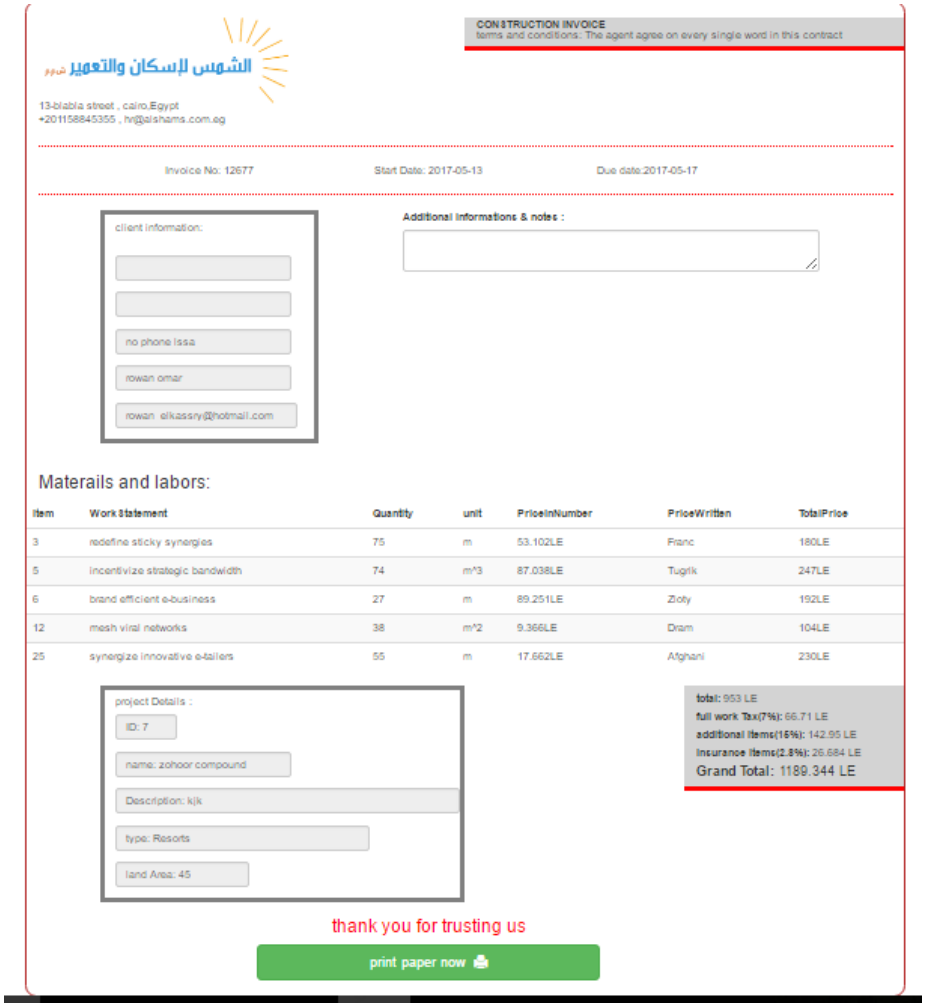

Figure 7: invoice

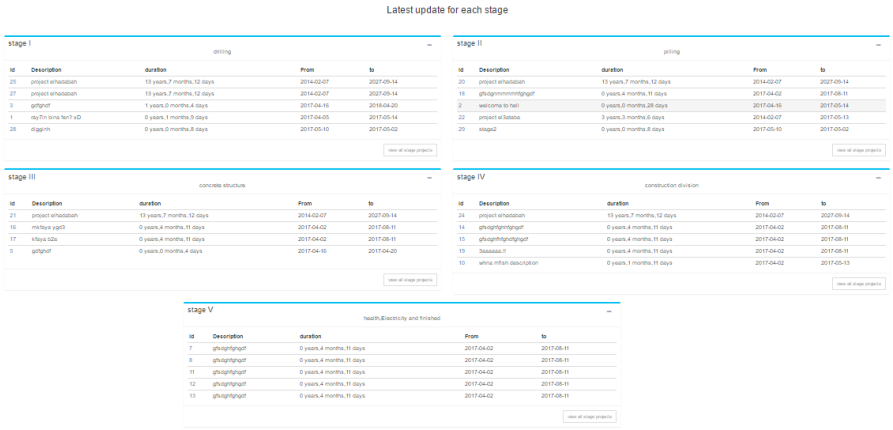

Figure 8: view stages

Figure 9: Add stages

	Assoc ID	Functional Requirment	logged in	registered	Project is applied by costomer	Invoices are listed by accountant	Commercial and Industrial Tax is calculated by accountant (permission)	user access is controlled by admin (permission)	User added by admin	User is updated by admin	user is deleted by admin	URLs are added by admin	user types are added by admin	project types are added by admin	contract types are added by admin	database orders are viewed by admin	Invoices are created by admin	attributes are added to form by admin	Materials are viewed by admin	forms are created by admin	user is search ed by admin	user is deleted by admin	project's timeplan stages are monitored by project manager
001	1.1.1	login		√					√														
002	2.2.2	register																					
003	3.3.3	Customer will apply for a new project	√																	√			
004	4.4.4	Accounting staff can list Invoices	√														√						
005	5.5.5	System shall Calculate Commercial and	√			√																	
006	6.6.6	Admin can control user accesspermissions	√									√	√							√			
007	7.7.7	Admin can add User	√										√										
008	8.8.8	Admin can Update User	√	√																			
009	9.9.9	Admin can delete User	√	√																			
010	10.10.10	Admin can add new URLs	√																				
011	11.11.11	Admin can add new user types	√																				
012	12.12.12	Admin can add new project types	√																				
013	13.13.13	Admin can add new contract types	√																				
014	14.14.14	Admin can view dashboards	√	√	√																		
015	15.15.15	Admin can view Invoices	√															√					
016	16.16.16	Admin can add attributes to forms	√																	√			
017	17.17.17	Admin can view Materials	√	√					√														
018	18.18.18	Admin can create forms	√									√											
019	19.19.19	Admin can search user	√	√					√														
020	20.20.20	Admin can delete User	√	√					√														
021	21.21.21	Project manager can monitor project's timeplan	√																	√			
		Technical office can																					

Figure 10: Traceability Matrix

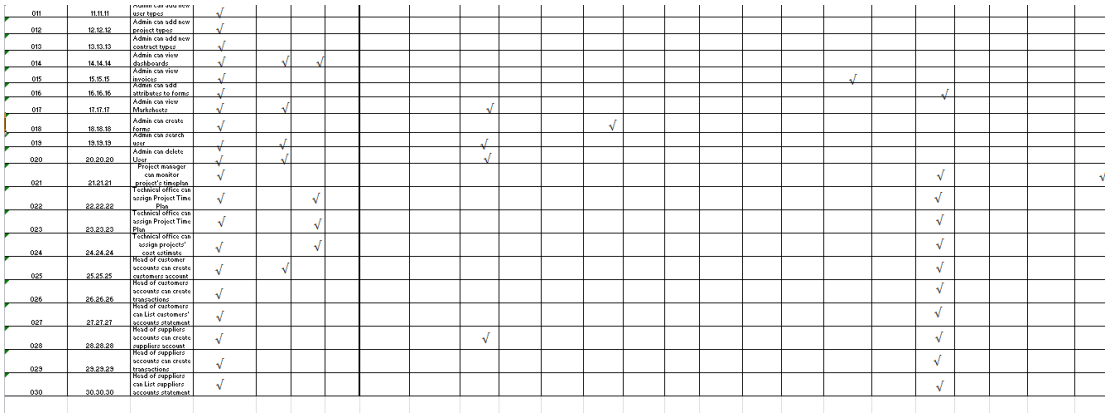

Figure 11: Traceability Matrix

Output: Alert Invalid Input (Date expired).

6.5 Add Contract

Happy Scenario:
Input: Choose Customer name and enter contract data (Notebook Number, State Contract, type of Contract, History of edits, Public Conditions, Date of Initial Receipt, Date of Final Receipt, Total Transaction Value, Down payment Value, Down payment Ratio, Pricing No, Pricing Case, Notebook Pricing No, Price Offering No, Price Offering Case, Price Offering Notebook No, Number of payments, Quantity per each payment, Tender No, Tender Case, Tender Description, First Party Signature, Second Party Signature).
Output: Contract Added successfully.
Bad Scenario:
Input: Add data of an existing contract.
Output: Alert Contract already exists.

6.6 Create Transaction

Happy Scenario:
Input: Choose account name and add debit, credit values.
Output: Transaction added successfully.
Bad Scenario:
Input: Input a negative currency.
Output: Alert invalid input.

6.7 View Transactions Test Case 1

Happy Scenario:
Input: Start/End Dates, Account Name and Code.
Output: List of all transaction within this interval of time.
Bad Scenario:
Input: Entering an expired date.
Output: Error Invalid Input.

6.8 View Transactions Test Case 2

Bad Scenario:
Input: Entering a code that doesnt exist.
Output: Alert transaction not found.

7 References