HOMEWORK 3. OO SYSTEM DESIGN

Homework 3. is a group project and can't be accomplished individually.

Homework 3 is devoted to Design phase of SDLC, particularly:

1. Re-define Requirements and Use-case Scenario
2. Update/Renew Use Case Diagram
3. Preparation for Design Phase
4. Classes and Methods Design
5. Data Management Layer
6. User Interface Layer
7. Physical Architecture Layer

All these steps must be applied to the system that was assigned to your team earlier. This phase is continuation of work done in the framework of the Homework 2.

1. Re-define Requirements

Re-define requirements (for more details refer to HW-2) after clearly understanding primary and additional functionality of the system. Follow to the below syntax:

[Client | User | Customer | Manager | Admin | System | Device | etc.] [does | can do | is expected | is intended | is required | should be able | etc.] [authorize | search | communicate | see | request | select | etc.] [… | SUBJECT/OBJECT | …] [and | or] [….]

Write main Use-Case Scenario and at list one Alternative Scenario.

2. Update/Renew Use Case Diagram

Update/Renew Use Case Diagram better understanding the functionality based on re-defined requirements (no need for use case cards and descriptions).

3. Preparation for Design Phase:

Implement V&V: Internal consistency and External consistency

- Functional Model V&V
- Structural Model V&V
- Behavioral Model V&V

Shortly describe how do you ensure V&V clearly separating each model (not more than two pages).

While moving from Analysis Model to Design Model (phase) which of following elements of Design Model you implemented (if any of them is not applied, explain why)?

- Factoring
- Partitioning
- Layers

4. Classes and Methods Design

All classes of the system included during Analysis Phase of SDLC should be designed and mainly implemented using particular Object-Oriented Programming Language that is intended to be used to development of the system. It is strongly recommended to use Java, otherwise, team should explain why it is necessary to use another language and what kind of advantages it brings in terms of OOAD.

WebPage.java

```java
public class WebPage {
    String url;

    public WebPage(String u) {
        this.url = u;
    }

    public void setURL(String u){
        this.url = u;
    }
    public String getURL(){
        return this.url;
    }
}
```

Repository.java

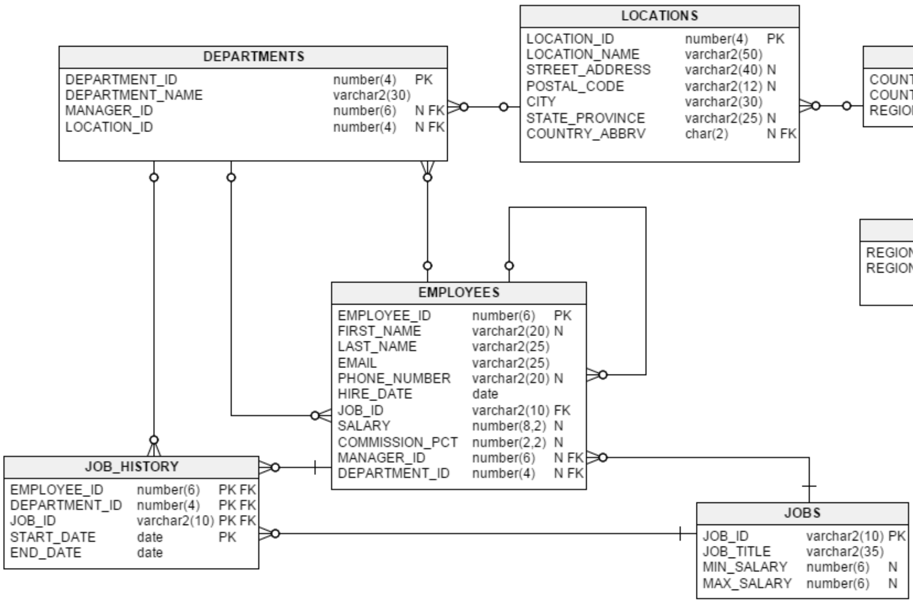
```java
public class Repository {
    private String login;
    private String password;

    File dir;
    FileWriter fstream;
    BufferedWriter out;

    public void storeContent(Document doc) {
        try {
                ...
        } catch (IOException e) {
            e.printStackTrace();
        }

    }
}
```

All classes with their attributes and methods should be designed in accordance to the Class Diagram. The Class Diagram at the Design Phase may be slightly different (thanks to V&V) from initial Class Diagram that was at the end of Analysis Phase. All Object-Oriented concepts must be implemented and clearly seeing in design of classes. As an output of this stage you have to have number of files (let say Java files) equal to number of classes in Class Diagram (file per class). See example above.

If number of implemented classes (files) is different from Class Diagram, you should explain which of them are not implemented and why. At this stage classes maybe incomplete without real implementation of methods having just signatures and partial implementation (or no implementation).

5. Data Management Layer

Entity Relation Diagram (ERD) – ERD is in some level similar to Class Diagram, but it doesn’t include methods. ERD diagram reflects real Data Model (Structure of Database) or Data Management Layer of the system. Example:


LOCATIONS
LOCATION_ID number(4) PK
LOCATION_NAME varchar2(50)
STREET_ADDRESS varchar2(40) N
POSTAL_CODE varchar2(12) N
CITY varchar2(30)
STATE_PROVINCE varchar2(25) N
COUNTRY_ABBRV char(2) N FK
DEPARTMENTS
DEPARTMENT_ID number(4) PK
DEPARTMENT_NAME varchar2(30)
MANAGER_ID number(6) N FK
LOCATION_ID number(4) N FK
COUNTRIES
COUNTRY_ABBRV char(2) PK
COUNTRY_NAME varchar2(40)
REGION_ID number N FK
REGIONS
REGION_ID number PK
REGION_NAME varchar2(25)
EMPLOYEES
EMPLOYEE_ID number(6) PK
FIRST_NAME varchar2(20) N
LAST_NAME varchar2(25)
EMAIL varchar2(25)
PHONE_NUMBER varchar2(20) N
HIRE_DATE date
JOB_ID varchar2(10) FK
SALARY number(8,2) N
COMMISSION_PCT number(2,2) N
MANAGER_ID number(6) N FK
DEPARTMENT_ID number(4) N FK
JOB_HISTORY
EMPLOYEE_ID number(6) PK FK
DEPARTMENT_ID number(4) PK FK
JOB_ID varchar2(10) PK FK
START_DATE date PK
END_DATE date
JOBS
JOB_ID varchar2(10) PK
JOB_TITLE varchar2(35)
MIN_SALARY number(6) N
MAX_SALARY number(6) N

6. User Interface Layer

Prepare the prototype of User Interface taking into account principles below. You are expected to provide at least 3 different UIs for different modules or sub-modules of the system under development. To create the prototype of UIs, you can use any IDE with GUI library of visual components, web user interface (WUI) or just graphical editor.

- Layout
- Content Awareness
- Aesthetics
- User Experience
- Consistency
- Minimal User Effort

7. Physical Architecture Layer

Describe how the system will function architecturally: 1-tier architecture, 2-tier architecture (Client-Server), 3-tier architecture (SOA, CORBA), etc.
Type of environment the system will work in: desktop, web app, mobile app.
Which platform the system is designed for: Windows, Linux, Mac, Android (mod), IOS (mob).
What kind of hardware and software the system will use? Which communication protocols will be utilized for interaction between components/sub-systems/tiers?
Additionally, draw the Deployment Diagram of the system.

Crucial Information

Important Recommendations: While working for this homework you are expected to work intensively with Internet resources collecting more information about similar software projects. Also, carefully review Part III (Design Modeling) of primary book and Section III.

Application (Chapters 8-12) of the (2nd recommended) book Object-Oriented Analysis and Design with Applications, 3rd Edition, Grady Booch.

Deliverables: The result is expected to be submitted as a single PDF Document that includes all sections of assignment (document must include all of five mentioned sections). As a cover page use "***cover-page.docx***" document available in course content. Single Java file or number of files with code in Java language same with number of classes and PDF document should be submitted as a single .ZIP file through Blackboard.

Deadline for submission: Sunday, 08th May, 2020 at 23:59 through Blackboard

Evaluation: Your work will be evaluated based on:

1. Completeness, relevance and accuracy of document
2. Appropriateness of classes to main services of the system and to Class Diagram
3. Correspondence of all between Analysis and Design phases
4. Overall quality of System Design

Completeness Checklist for HW-3

☐ Requirements
☐ Use Case Diagram
☐ V&V of Models Consistency
☐ Design Model Optimization (Factoring, Partitions, Layers)
☐ Package Diagram
☐ Class Diagram
☐ Java Code
☐ Deployment Diagram

Evaluation Rubrics

Criteria / Weight	Exemplary 5	Satisfactory 3	Need Improvement 1
Professionalism (x2)	Document is composed professionally, all parts are consistent is terms of notations and system itself, design elements and phases are clearly communicated and supported.	Document is composed satisfactory, all parts are partially consistent, design elements and phases are not enough communicated and supported.	Document is composed purely and there is no consistency.
Preparation for Design Phase (x3)	Preparation for design is well done approving V&V in all models. Optimization of Design is robust implementing Factoring, Partitioning, Layers very well.	Preparation for design is satisfactory. V&V is not clear and some models are questionable. Factoring, Partitioning, Layers are not enough clear.	Preparation for design is done purely and some elements are not implemented.

Clarity of Diagrams and Correct Use of Notation (x2)	Diagrams are well labeled and at an appropriate level of abstraction so that stakeholders familiar with the problem domain could readily understand them. All notation used in the diagrams is appropriate to the diagram type and is used correctly.	Diagrams are mostly well Labeled. Diagrams are generally at an appropriate level of abstraction, though a stakeholder familiar with the problem domain might need some guidance to understand them. All notation used in the diagrams is appropriate to the diagram type.	Labels are often cryptic or abstraction is used to the point that the actual analysis and design implications would be obscured to all but an expert in both the notation and the domain. Diagrams use notation inappropriate to the diagram or contain errors
Codding Style and Consistency with Design (x4)	Code is clear and well documented with consistent and appropriate naming and formatting. Code for the system is consistent with Diagrams, both structurally and behaviorally.	Code is mostly clear and documented. The majority of identifiers are well named and the formatting is mostly consistent. Code for the system is mostly consistent with Diagrams.	Code is often unclear or undocumented. Obscure or terse identifiers are the norm. Code for the system is mostly not consistent with Diagrams.
Data Managements (x3)	Data Modelling is done perfectly and fully meet requirements of the system. DM follows to all 3 NF. DAM classes are exist in Class Diagram and Code;	Data Modelling is done mostly well and partially meet requirements of the system. DM partially follows to NF. DAM classes are partially exist in Class Diagram and Code;	Data Modelling is done purely. DAM classes are missing.
User Interface and Physical Architecture (x2)	User Interface is fully consistent with system's functionality and follows to all UI requirements. Physical Architecture is relevant for the system.	User Interface is mostly consistent with system's functionality and follows to some of UI requirements. Physical Architecture is mostly relevant for the system.	User Interface is not consistent with system's functionality and doesn't follow to UI requirements. Physical Architecture is mostly not relevant for the system or totally wrong.
Overall effectiveness of Analysis and Design Models (x4)	Analysis and Design convey all important elements, constructs, and behaviors. It demonstrates a deep understanding of the solution to the problem.	Analysis and Design convey many key elements, constructs, and behaviors. Some situations might be treated in an unusual manner, and understanding of most of aspects of the system is demonstrated.	Analysis and Design minimally convey key elements, constructs, and behaviors. It shows a superficial understanding of the problem and its solution

Wishing you success,
Dr. Abzetdin Adamov