

SEP Online

System Design

Version 1.0

11 April 2018

Team Members:

Oğuzhan Ulusoy

Ayşenur Bayram

Tahir Tekizbaş

Didar Turanlı

Dilara Turanlı

Prepared for
SE302 Software Engineering Practice



IŞIK UNIVERSITY
COMPUTER
SCIENCE AND
ENGINEERING

Table of Contents

1.	Introduction.....	1
1.1.	Purpose of the System	1
1.2.	Design Goals.....	1
1.3.	Definitions, Acronyms, and Abbreviations.....	1
1.4.	References	2
2.	Current Software Architecture.....	2
3.	Proposed Software Architecture	2
3.1.	Overview	3
3.2.	System Decomposition.....	3
3.3.	Hardware Software Mapping.....	6
3.4.	Persistent Data Management	6
3.5.	Access Control and Security.....	8
3.6.	Global Software Control.....	8
3.7.	Boundary Conditions.....	8
4.	Subsystem Services.....	10
5.	References.....	12

SYSTEM DESIGN DOCUMENT[1]

1. Introduction

Design is the abstraction of the solution of a problem in general terms and it is also a visual transfer of a developed application. Successful completion of the design allows to save the time spent in the application part.

1.1. Purpose of the System

The main purpose of the system is to provide quality, fast and secure service to users. The system has some useful features that can help user to display items as periodically and find local market easily. Users can find items' information with just a few clicks. Users do not have to register to search and view operation, however registration is necessary to write blog, and add comment. Furthermore, the system is completely free.

1.2. Design Goals

Usability

The SEP Online should have three different navigation menus for all actor. Normal user should use standard navigation menu. Author should have a navigation menu that is including standard navigation menu and “write blog” link. Admin should have a navigation menu that is including standard navigation menu and “admin panel” link. Each page should contain same header, footer, head, content blocks. Almost every page (both static and dynamic) should have same body page. Profile page, register, login is basic-level membership. The content should be shown as quite simple. There should no guide for users, because SEP Online should be created easier for users.

Reliability

The SEP Online should be provided to access for all members by secure. Unregistered visitor should be able to do some stuff. Yet, registered visitor, which are author and normal user, should be able to do some stuff more different than unregistered visitor. Logging to the web-site should be provided with unique e-mails and password that are appropriate for password criteria. The passwords should be hit as hashed password in database. The membership information should be private and should not be shared with anyone. Session and PHP Data Object are used to provide reliability.

Performance

The SEP Online should be responsive in different devices. SEP Online should be shown in mobile phones, tablet and computers. SEP Online should be running in more than one computer simultaneously, and access should be guaranteed. The web-site is going to be

SEP Online

dynamic content, so there should not be complicated queries in back end to not decrease performance.

Supportability

The system should be managed by admin. The system should be supported on different browser and be independent by hardware. In FTP side, there are going to be some content components (*i.e. header, head, script, footer, and so on*) to maintain the system easily, rapidly and properly.

Implementation

There are no constraints on the hardware platform. There are no constraints imposed by the maintenance team. There are no constraints imposed by the testing team. The design methodology is MVC (model-view-controller). We will check input over controller, our model is satisfied object relation model over SQL. PHP language is used in this system. HTML5, CSS and Bootstrap (*briefly*) should be used to implement.

Interface

The system should not interact with any existing system. The system should be able to use by a user. The user should be connected to the network to use the features of the system.

Packaging

Admin should install the system. Also, the system is a web site so the site should uploaded on the server. There should no time constraints on the installation.

Legal

The SEP Online does not use any license or licensed stuff. It is a social media network.

1.3. Definitions, Acronyms, and Abbreviations

PHP	: Hypertext Preprocessor
SDD	: System Design Document
ADDOT	: Online bazaar system (SEP Online)

1.4. References

References to existing systems, etc.

2. Current Software Architecture

There are some websites that are related to local producers. These websites are to provide shopping for consumer. The web sites mentioned are TazeDirekt.com[2], TazeMasa.com[3]

and MemlekettenGelsin.com[4]. There are some websites that are related to healthy life including blog-posts. One of the web sites mentioned is vitaminrehberi.org[5]. There is no web-site which has goals like the we proposed system. These systems use marketing strategy to show own system with some design patterns that make the job easier. In terms of, our system will include some parts of the existing systems.

3. Proposed Software Architecture

We suppose a system that contains strategy design pattern. According to this edesign pattern, we do not clearly implementation. However, when the strategy design pattern leads to our project. Accordingly, static contents will be implemented in *contents/statics* folder. The files in this folder, do never edit while implementation period. Also, the pages (*visible ones in menu navigator bar*) will be implemented in main folder and these files use some contents from *contents/statics* folder-way with include command in PHP. Thus, maintenance and operate become easier and proper.

Also, we suppose a system that has been implemented partly by Service Oriented Architecture (*SOA*) or Service Oriented Programming (*SOP*). According this Service Oriented Programming, when we implement our project, all of the functions are called from one source that is called manage file. It is addressed to *services/manage.php*. This file contains all functions that are called web-service. All methods will be here.

In addition to these, manage file includes controllers. These controllers do controlling of inputs and so on.

The system is not objective-oriented. Therefore, there is no model. However, database of our system will be query-based. With completely, this is back-end of the project.

In front-end, we have fundamental pages. These pages use to statics sections with include command from a folder-way. Also, these pages call related functions from services folder. So that, our views are generated by this.

3.1. Overview

The system is user friendly system and all user can use the system easily. The system is made of four interfaces.

User Interface: It provides services for all users for common interfaces like a login form, password change form, membership profile and so on.

Admin Interface: It provides services to manage for system administrator. They are likely forms or lists such as add operations (add new item, add new local market and so on) are forms, the others are lists (change item, change local market, delete item, delete local market and so on).

Visitor Interface: It provides services for visitor, such as displaying local markets and so on.

Registered User Interface: It provides services for author, such as writing blog post.

The system is divided into sub-systems. These sub-system makes our work-breakdown structure. They are similarly to tasks.

Add Location Subsystem: It provides authority for system administrator to add new location with own attributes into the system.

Delete Location Subsystem: It provides authority for system administrator to delete location from the system.

Add Local Market Subsystem: It provides authority for system administrator to add local market with own attributes into the system.

Delete Local Market Subsystem: It provides authority for system administrator to delete local market from the system.

Display Local Market Subsystem: It provides service for all actors on system to display local market with own attributes.

Change Local Market Subsystem: It provides authority for system administrator to change local market with own attributes on the system.

Add Item Subsystem: It provides authority for system administrator to add item with own attributes into the system.

Delete Item Subsystem: It provides authority for system administrator to delete item from the system.

Change Item Subsystem: It provides authority for system administrator to change item with own attributes on the system.

Display Item Subsystem: It provides service for all actors on system to display item with own attributes.

Add Blog Subsystem: It provides authority for author to add blog with own attributes into the system.

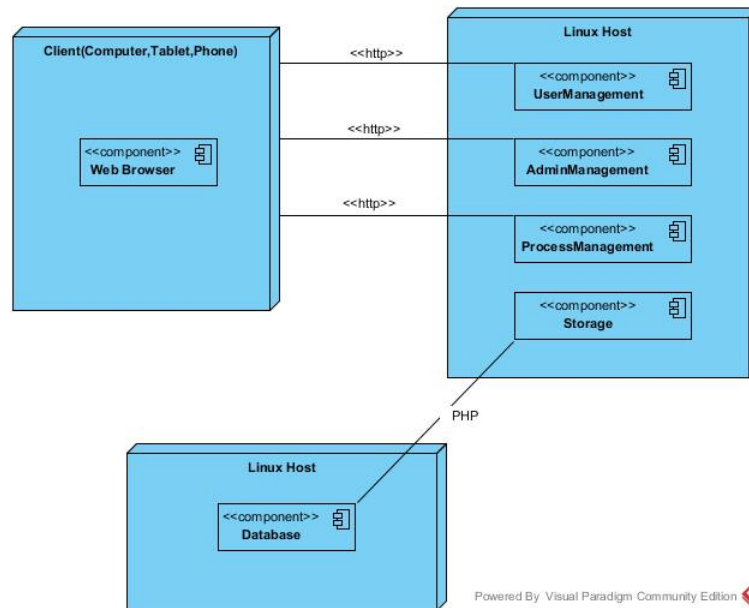
Delete Blog Subsystem: It provides authority for system administrator and author to delete blog from the system.

Suggest Item Subsystem: It provides service for members to suggest an item.

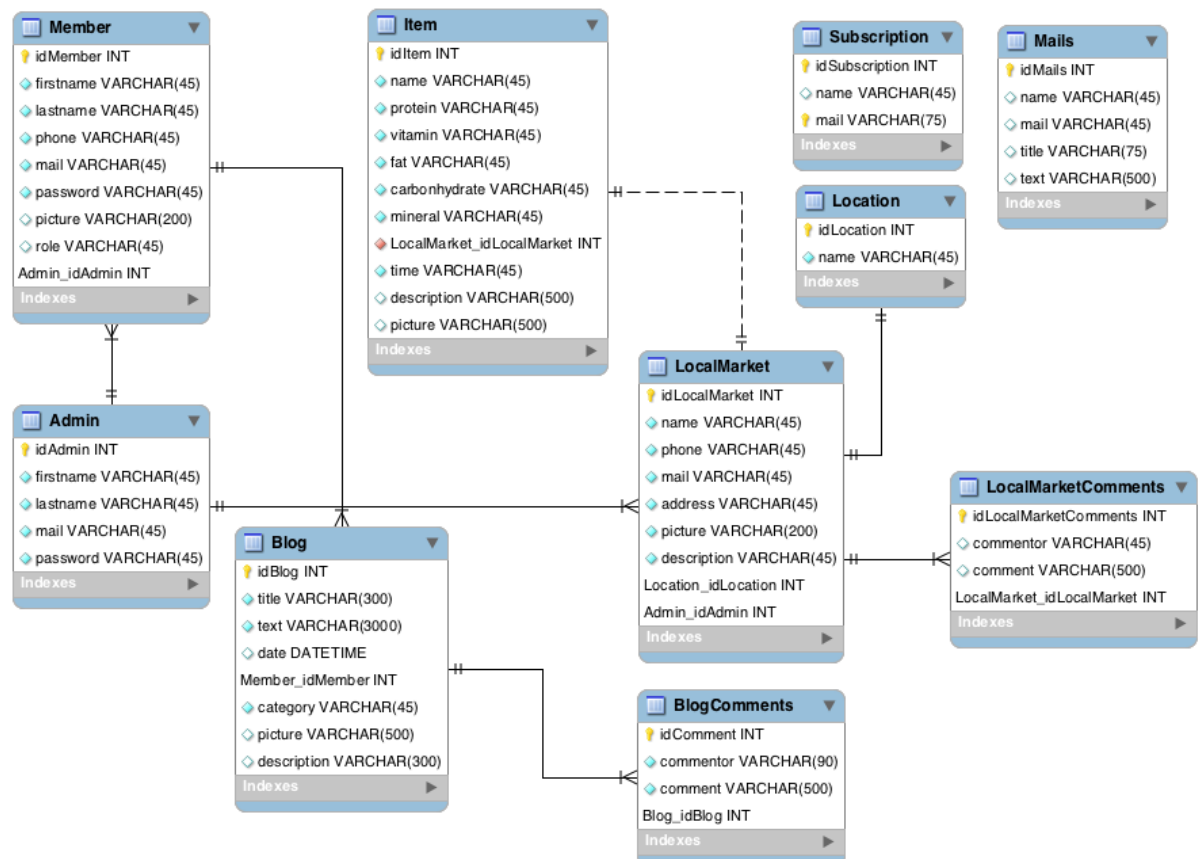
Profile Subsystem: It provides service for members to display profile with own attributes.

3.3. Hardware Software Mapping

Hardware-Software Mapping has been created by third party software (Visual Paradigm).



3.4. Persistent Data Management



SEP Online

Member

Member table stores all information about members. The table has nine different columns. These are idMember, firstname, lastname, phone, mail, password, picture, role and idAdmin. idMember is an integer unique primary key, it is generated by system. Firstname, lastname, mail are string data. Mail is also primary key and should be unique. Password is a string, at registration phase, there are some rules to obtain password. idAdmin is for authors. Normal users have 0-value as default for idAdmin.

Admin

Admin table stores all information about admins. The table has nine different columns. These are idMember, firstname, lastname, phone, mail, password, picture, role and idAdmin. idMember is an integer unique primary key, it is generated by system. Firstname, lastname, mail are string data. Mail is also primary key and should be unique.

Item

Item table stores all information about items. The table has eleven different columns. These are idItem, name, protein, carbohydrate, fat, mineral, vitamin, idLocalMarket, time, description, picture. Except idItem and idLocalMarket, all of them are string.

Local Market

Local Market table stores all information about local markets. The table has eleven different columns. These are idLocalMarket, name, phone, mail, address, picture, description, idLocation, idAdmin. Except idLocalMarket, idLocation and idAdmin, all of them are string.

Location

Location table stores all information about locations. The table has two different columns. These are idLocation and name. idLocation is integer, name is string.

Blog

Blog table stores all information about blogs. The table has eight different columns. These are idBlog, title, text, date, idMember, category, picture and description. idBlog and idMember are integer type. Date is date type. Rest of columns are string types.

Blog Comments

Blog Comments table stores all information about blog comments. The table has four different columns. These are idComment, commentor, comment, idBlog. idComment and idBlog are integer types. Comment and commentor are string types.

Market Comments

Market Comments table stores all information about market comments. The table has four different columns. These are idComment, commentor, comment, idLocalMarket. idComment and idLocalMarket are integer types. Comment and commentor are string types.

3.5. Access Control and Security

Describe the user model of the system in terms of an access matrix. This section also describes security issues, such as the selection of an authentication mechanism, the use of encryption, and the management of keys.

SEP Online is a web-based application running with query. It is not created with object-oriented programming, therefore there is no classes. It is classical web-based application. It is a multiuser system which means that there are many actors more than one actor. The mentioned actors are four people in totally that are system administrator, author, normal user and visitor. However, we focus on major two types that are system administrator and normal user. Because, the difference between author and normal user is to write blog. Visitor behaves like not registered anybody. In terms of, focusing on system administrator and normal user will be explained all scenarios clearly.

If there has been objective oriented architecture, there should have implemented some classes that are organized by actors. In this project, we are setting up all encapsulation through the role in database. When a registered person tries to log in to the system, if there is no problem, she will be redirected to main page as her role. Actually, she accesses some services by visible.

The criteria of access to the system depends on unique defined e-mail and password at registration phase.

3.6. Global Software Control

The system controls the authentication with defined unique mail and password. The synchronization is satisfied among queries on server. Thus, dynamic content will be displayed.

3.7. Boundary Conditions

SEP Online is activated by the system admin calling the ‘Initialize System’ use case. Once the initialization of the system is completed, the users can are register onto system. Moreover, the system admin activates the server by calling the ‘Start Server’ use case. Furthermore, when the system is brought from non-initialized state to steady-state, the system is opened and free for the system users (User, Visitor and Admin) to login and perform their tasks. While the users of the system are online and perform their tasks to accomplish, there might be some errors occurring during the transaction processes of the tasks that performed by the users.

The mentioned errors could be originated from attempting to login with invalid credentials, trying to update the personal information (which are necessary to be filled) with empty information. All these exceptions are get caught by the system and handled properly.

SEP Online

For instance, for attempting to login with invalid credentials error, the use case name 'Invalid Credentials' is called to handle the exception. When the 'Invalid Credentials' use case is invoked, the system has already checked the user's mail and password that if the user mail and password matches with the mail and password in the database of the system. The system realizes that they do not match with each other, then in the process of the 'Invalid Credentials' use case, the user, who tries to login, is being informed with a proper message indicates that "E-Mail or Password is Incorrect". Thus, the system lets user to know that the credentials that the user uses to login in not correct so that the user can realize the situation and try again with the correct one. For trying to update the personal information (which are necessary to be filled) with empty information exception, the exception occurs when the current logged user tries to update his/her personal information but supply empty fields which are not supposed to be empty. Then SEP Online performs a checking to make sure that the necessary information fields are given, and the information fields are proper to be updated. However, the system realizes that the necessary fields were not supplied, so the use case name 'Missing Fields' is invoked to handle the exception. When the 'Missing Fields' use case is invoked, a proper message (the message indicates the necessary fields which are not supposed to be empty but left empty by the user) is display to the logged user to fill the necessary fields to accomplish the update task.

When the system is terminated by system admin invoking the 'Shutdown Server' use case. All users that currently logged in to the system will be disconnected. No user can login to the system until the system is initialized by the admin again.

The use cases mentioned above, namely 'Start Server', 'Shutdown Server' can be seen below:

<i>Use case name:</i>	InitializeSystem
<i>Participant actors:</i>	Initiated by Admin
<i>Flow of events:</i>	1. Upon successful login, the Admin executes the commands to register the client and guest information.
<i>Entry Condition:</i>	The Admin logs into the database server that the SEP Online uses.
<i>Exit Condition:</i>	Initialization of the system is completed and the all user and item information of are registered onto the database server.

<i>Use case name:</i>	StartServer
<i>Participant actors:</i>	Initiated by Admin
<i>Flow of events:</i>	1. Upon successful login, the Admin executes the command that is used for the start the server.
<i>Entry Condition:</i>	The Admin logs into the server machine that the SEP Online services built on.
<i>Exit Condition:</i>	SEP Online services are available and waits for connections from user of the SEP Online.

<i>Use case name:</i>	ShutdownServer
<i>Participant actors:</i>	Initiated by Admin
<i>Flow of events:</i>	1. Upon successful login, the Admin executes the command that is used for the shut down the server.
<i>Entry Condition:</i>	The Admin logs into the server machine that the SEP Online services built on.
<i>Exit Condition:</i>	SEP Online services are unavailable and connections which were already activated are disabled.

4. Subsystem Services

During the subsystem decomposition of SEP Online system, we divide the system into smaller subsystems with strong coherence. The different subsystems should have a loose coupling.

The subsystem separation shows the entities of the following subsystems:

- I. User management subsystems
- II. Admin management subsystems
- III. Process management subsystems
- IV. Database subsystems

User management subsystems

This subsystem responsible for managing different users of the system by taking care of login information of different users. It manages the username and password of all users of the system.

Operations provided by this subsystem are:

Login()
Logout()

Admin Management Subsystem

This subsystem responsible for managing user accounts and local market. It provides function for opening an account, updating an account and closing an account. Admin is the actor who communicates with this subsystem. This subsystem uses user management subsystems for authenticating the admin, user their information and voyages information.

The operations provided by admin management subsystems are:

Login()
AddLocalMarket()
DeleteLocalMarket()
AddItem()
DeleteItem()
DeleteBlog()
DeleteComment()
ForgetMyPassword()
ChangeLocalMarket()
ChangeItem()
ChangeProfile()
DeleteMember()
Logout()

Process Management Subsystems

This subsystem is responsible for managing the process. This provides all functions for managing local market to details, view, post, comment and other things.

User performed by this subsystem are:

Login()
DisplayProfile()

SEP Online

DisplayBlog()

WriteBlog()

WriteComment()

ChangeProfile()

SuggestItem()

Search()

DisplayLocalMarket()

ChangePassword()

ForgetPassword()

DisplayItem()

Logout()

Visitor performed by this subsystem are:

Register()

DisplayLocalMarket()

DisplayItem()

DisplayBlog()

Database Subsystems

The database subsystem will be implemented by relational database management system used to store all data.

Admin Interface Subsystems

This subsystem responsible for managing user and visitor information. It provides function for local market and post information. Admin is the actor who communicates with this subsystem. This subsystem uses user management subsystems for authenticating the admin user information.

User Interface Subsystems

This subsystem in charge of managing the process. This ensures to edit profile, suggest item, display profile, post and local market, local market and item.

5. References

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on [1]).

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.