<WWDS>

System Design

<1.0>

<8.12.2017>

<Serdar Ünlüsoy
Mercan Gültekin
Erdal Yurtseven
Muhammed Ali Altınel>

Prepared for
SE301 Software Engineering

IŞIK UNIVERSITY
COMPUTER
SCIENCE AND
ENGINEERING

Table of Contents

SYSTEM DESIGN DOCUMENT 1

1. Introduction

1.1 Purpose of the System

WWD is the system that designed for the Users who would like to donate to one or more projects and to start a project or more projects. It is designed to accomplish signup and user transactions.

After the Users login into the system, they are able to use operations that are offered to their accounts. Therefore, donation and starting a project can be performed by the Users.

The system interface is designed as user friendly. As all transactions that are related to users can be performed explicitly in steps, users can operate actions without any complication.

1.2 Design Goals

- ★ WWD system is designed to be used easily. Any user who knows their Username and password can use the system and perform operations.
- ★ User interface should be designed for any user to accomplish their operations with basic, simple and understandable steps. In addition to the registered users, visitor users can use search and information parts.
- ★ The system should not allow unauthorized users to access user information and operations. To maintain the security of the system, the system is designed as not accessible to operations that are offered to users without login into the accounts by exclusive Username and password.
- ★ The system can handle expected exceptions such as invalid input by showing the user a related message and asking the user to enter the input again.
- ★ To avoid any discomfort, the system should keep the persistent data correctly.
- ★ The system architecture is designed to be readable by other programmers. Because of the subsystem decomposition, it is easy to adapt changes and new functions without the much need of recompile the code.

1.3 Definitions, Acronyms, and Abbreviations

API: An application program interface (API) is code that allows two software programs to communicate with each other

SDD: System Design Document

SYSTEM: Any interactions performed by the application are considered to be performed by the system.

SQL: Structured Query Language

MSSQL: Short for Microsoft SQL Server. It is a relational web hosting database that is used to store web site information.

MVC: The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller.

MODEL: A schematic description of a system that accounts for its known or inferred properties.

VIEW: A visual representation of a model which might.

HTTP: Hypertext Transfer Protocol

1.4 References

Our new system is described in the “Current Software Architecture” part of the SDD.

2. Current Software Architecture

Our system has 3-layered software architecture, which consist of *Model-View-Control* approach.

The model-view-controller (MVC) style decouples data access (entity objects) and data presentation (boundary objects).

Subsystems are classified into three different types:

– Model subsystems maintain domain knowledge, do not depend on any view or controller subsystem. (*Entity objects)

– View subsystems display it to the User (**Boundary objects*)

– Controller subsystems manage the sequence of interactions with the User. (**Control objects*)

A special case of the repository where Model implements the central data structure and Control objects dictate the control flow. MVC is well suited for interactive systems, especially when multiple views of the same model are needed.

MVC can be used for maintaining consistency across distributed data.

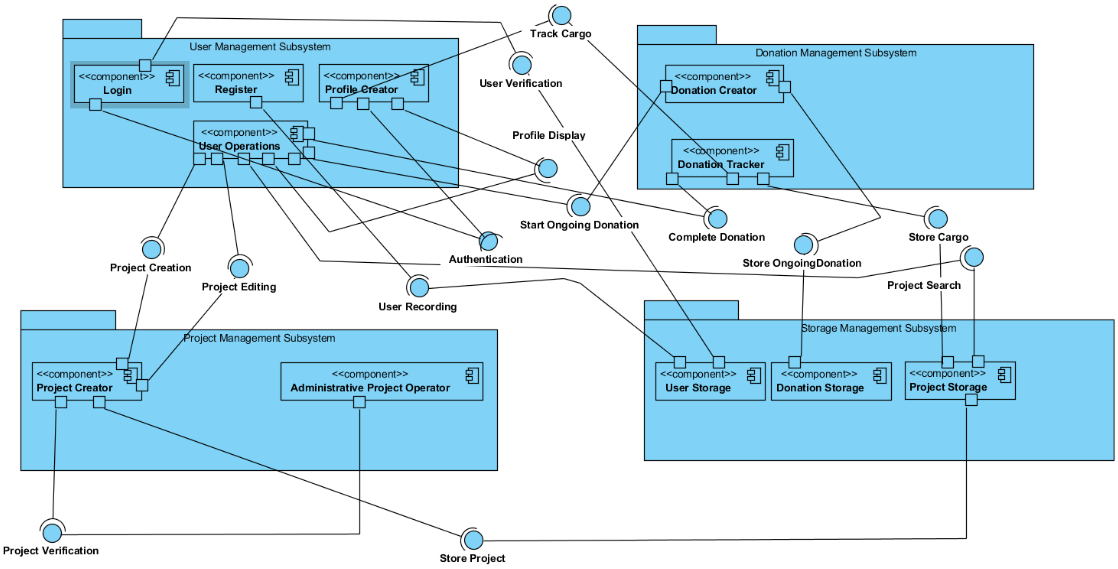
3. Proposed Software Architecture

3.1 Overview

In this section we will introduce a main structure of the system and describe the main aspects of the design how system will perform. This section also provides a short explanation about each component and process to be developed under the Worldwide Donation System project.

So, this section includes, decomposition of the main system into subsystem, how these subsystems connect with each other, data management, access control, control flow and boundary conditions of the system in case of errors and characteristics of the services provided by the subsystem.

3.2 System Decomposition


User Management Subsystem
<<component>> Login
<<component>> Register
<<component>> Profile Creator
<<component>> User Operations
Track Cargo
User Verification
Profile Display
Donation Management Subsystem
<<component>> Donation Creator
<<component>> Donation Tracker
Start Ongoing Donation
Complete Donation
Store Cargo
Authentication
Store OngoingDonation
Project Search
Project Creation
Project Editing
User Recording
Project Management Subsystem
<<component>> Project Creator
<<component>> Administrative Project Operator
Storage Management Subsystem
<<component>> User Storage
<<component>> Donation Storage
<<component>> Project Storage
Project Verification
Store Project

User Management Subsystem:

Basic operations done in this subsystem for supporting functions that users perform. This subsystem also provides authentication service for any other subsystem that needs authentication for an operation.

Donation Management Subsystem:

Provides sundry services for operations that are related with donations.

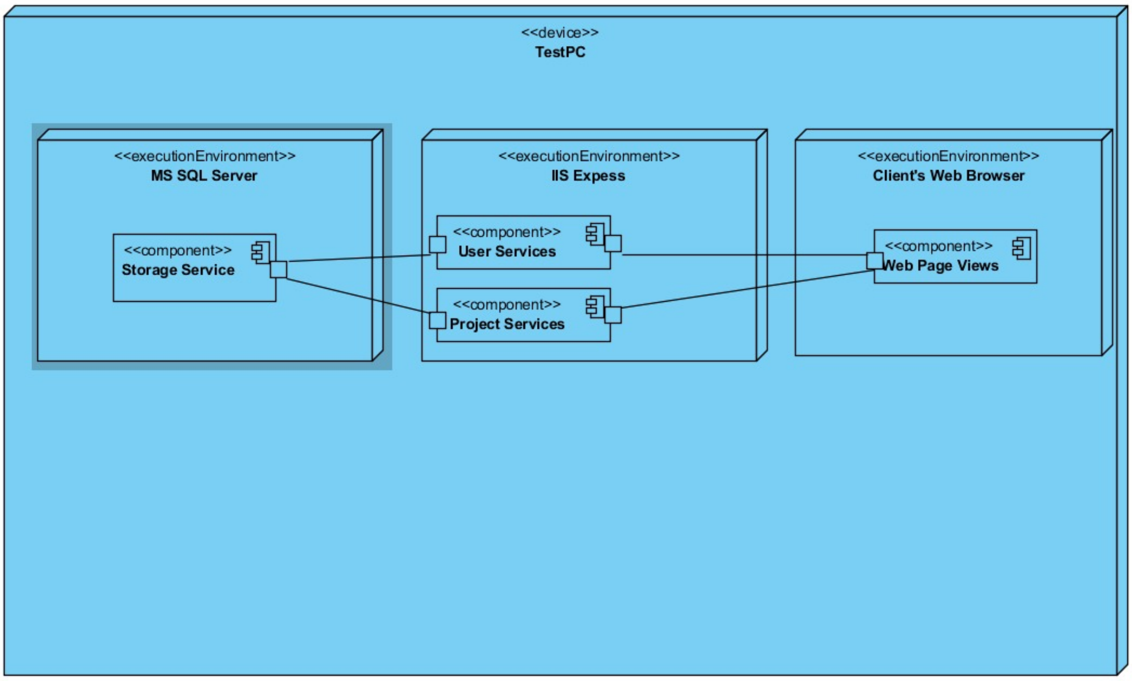
Project Management Subsystem:

Project processes are handled in this subsystem.

Storage Management Subsystem:

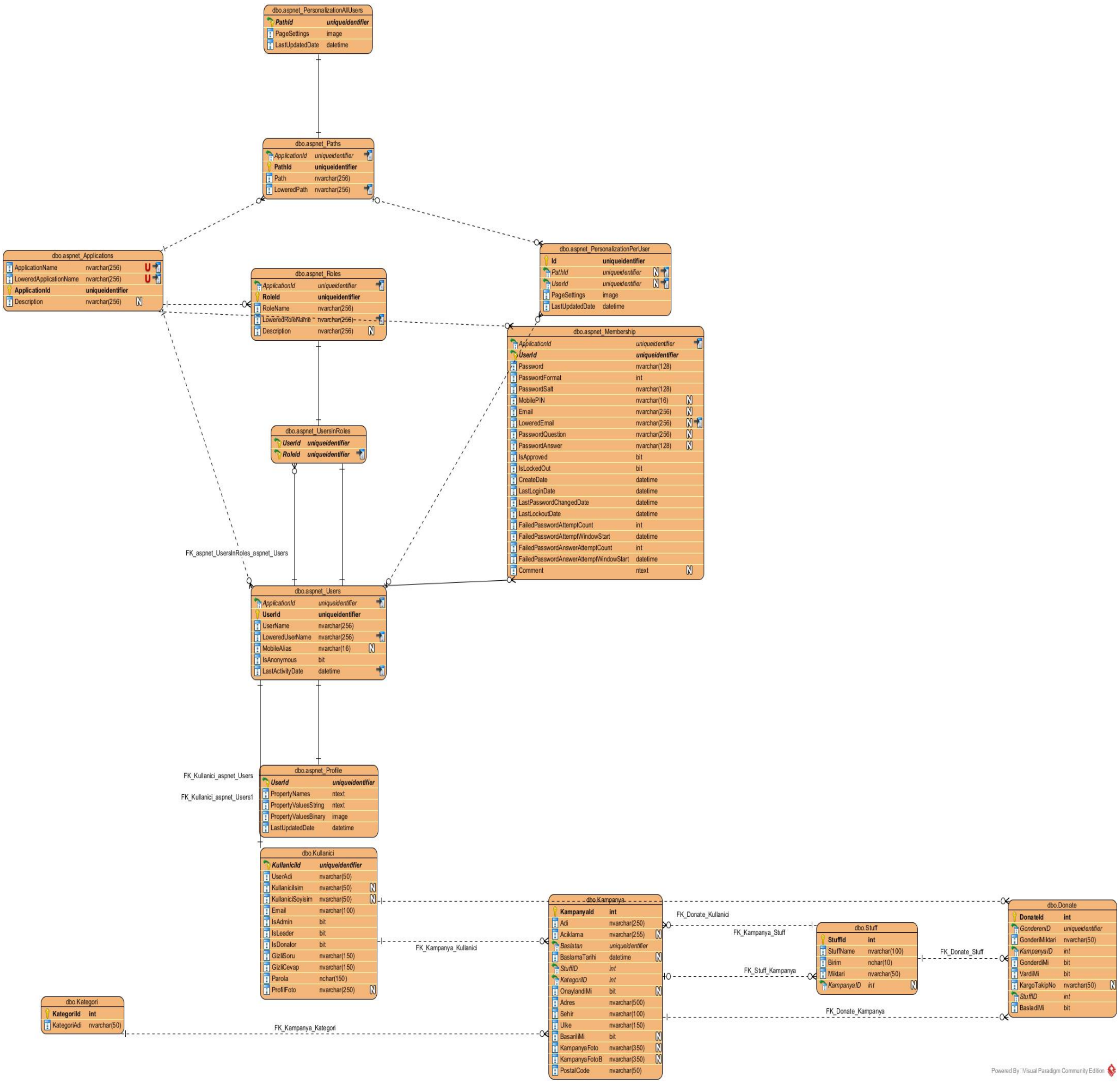
This subsystem is basically an abstraction layer for actual storage unit in order to make the system independent from database language changes.

3.3 Hardware Software Mapping



WWDS is a web site, and it does not need any installation to *User Machine*. To access the site a *User Machine* must have an internet connection, and a web browser such as Chrome (by Google). After these conditions provided, one can enter the URL address of the site and once the browser contacts the WWDS's server. Then, one will be able to access the WWDS's web site.

3.4 Persistent Data Management



The data classes support persistent management of data that is maintained on backend database. These classes include the SQL classes to let we manipulate persistent data stores through a standard SQL interface. Additionally, a set of classes called ADO. **NET** and entity framework allows us to manipulate persistent data. Persistent data outlive a single execution of the system. This is why WWDS must store its lifelong data. In order to keep its data, WWDS system must have a database and a database connection. With this connection database will be created, corresponding data will be stored.

Data which will be stored are the ones which are critical for us. Like the Users of the system. All Users have primarily information which is User ID and password. They must be kept in database but WWDS wants to store their Users' personal information such as E-mail, Name, Surname, Username. There is a dilemma in database management which is choosing memory or speed. Choosing speed will leads us to use more space, choosing memory will leads our system to run slower. Since we are aiming to use complex queries speed will be our priority.

“Relational Database” will be more useful for handling large amount of data in a database. Also, it has more privileges like we can use complex queries more efficient than the “Object-oriented Database”

3.5. Access Control and Security

In our project, WWDS, there are 3 types of actor which is Visitor, User and Admin. Visitors are the ones who are not logged into the WWDS and/or who has no information to log into the WWDS. Users are the ones who are able to log into the WWDS with their login information.

The implementation of the Access Matrix to the WWD system, adopts the idea of “Access Control List”. We identify (Actor, Operation) pair for each classes. By implementing this we are aiming for classes to check their own accesses as Bruegge and Dutoit mentions “In Access Control List, every time an object is accessed, its access list is checked for the corresponding actor and operation”

	PROJECT	DONATION	USER	STUFF

VISITOR	Search()
	DisplayProjectInformation()
			Signup()	

USER	DisplayProjectInformation()
	Search()
	StartProject()
	DisplayCargoListOfMyProject()
	EditMyActiveProject()
		Donate()
		CancelDonation(
		CompleteDonation()
		VerifyCargo()
		RejectCargo()
			Login()
			Logout()
			DisplayOngoingDonation()
			DisplayProfile()
			DisplayDonatedProjects()
			DisplayMyfinishedProjects()
			DisplayMyActiveProjects()
				InsertNewStuff()

ADMIN	DisplayProjectInformation()
	Search()
	VerifyProject()
	RejectProject()
	VerifyEdit()
	RejectEdit()
			Login()
			Logout()	

3.6. Global Software Control

In this project we will use the Model View Control approach. User enter his/her data.

Controller initiate model. After that, data gained from model is viewed in the WWDS system web site.

Since our system is interactive web site, it would be suitable to use MVC.

Also, Visual Studio which is the environment we work with is compatible with Model View Controller.

Donation Management Subsystem and Project Management Subsystem are very important for WWDS system.

In total, both of these subsystems contain services which make base of the WWDS system. For example, Donation Manage Subsystem provide service called Start Ongoing Donation which is a service about starting donation which is primary concern of the WWDS system. Project Management Subsystem provide a service called Project Creation service which is about creating new donation project. Further details about Project Management Subsystem and Donation Management Subsystem and their services will be discussed in Section 4.

User Management Subsystem handle User operations and more importantly synchronize with Donation Management Subsystem and Project Management Subsystem but User Management Subsystem synchronized with them separately because Donation Management Subsystem and Project Management Subsystemhave different purpose and both of them are independent subsystem.

User Management Subsystem depends on other subsystems very highly. That means User Management Subsystem have very high concurrency which will should make us cautious when we implement Donation Management Subsystem and implement Project Management Subsystem otherwise system would not work as way it should.

Storage Management Subsystem provide storage services to User Management Subsystem,

Donation Management Subsystem and Project Management Subsystem. That way, we manage to create another

synchronization. Storage Management Subsystem store information from every subsystem.

User Management Subsystem, Donation Management Subsystem and Project Management Subsystem dependent on

Storage Management Subsystem because Storage Management Subsystem deals with storage of entire information management of the

system which is incredibly important.

3.7 Boundary Conditions

Start Up: When the system is first deployed to a web server it is started by the developers of the system. By this operation system already has an administrative account added to database by manually. Client is given this admin account if requests. After these operations the system is started and user only has to visit the web page of the WWDS on his/her device.

Shut Down: When shutdown requested by admin, the server terminates all ongoing operations shuts down the system. No data gets corrupted. All critical and crucial data are stored in system database.

In WWDS, five major exceptions might be arisen:

1) Invalid input exception in which the input entered by the Users during transactions does not match the wanted conditions.

2) Non-existence of the input exception in which the input entered by the Users during transactions does not exist in the system.

3) Already exist/selected input exception in which the transaction performed by the Users has already performed.

4) Incomplete input exception in which the input entered by the Users during transactions is not sufficient to end the transaction successfully.

5) Server crashes exception in which WWDS server is unexpectedly terminated.

We decided to handle the first four, input related, exceptions as showing a message that explains the failure of the transaction to the Users. Then the system will ask the Users to enter the correct input to end the operation successfully.

4. Subsystem Services

We have four different subsystems mentioned in System Decomposition section. Each of subsystems contain component and component's service(s). In this section, Subsystems, their component, their services, which component provide which services and which component require which services briefly.

User Management Subsystem:

User Management Subsystem manages the users in the system whether if the user is registered user or nonregistered user. This subsystem contains components such as Login, Register, Profile Creator and User Operations. Components of each subsystem's services are described below.

1) Login:

This component logs the User in.

1) Authentication: This service is provided by login component and required by Profile Creator subsystem. Users of WWDS system login process or authentication process handled by Authentication service.

2) Register:

This component register User to WWDS after User registration information given by user is stored in the WWDS database.

3) Profile Creator:

Profile Creator component display information about User profile

1) Profile Display: Profile Creator component load Profile Page and displays it to User. Profile Creator component provide Profile Display service and User Operations component require this service.

Project Management Subsystem:

Project Management Subsystem concerns with creating project and editing project. This subsystem also requires services such as Project Verification and Store Project.

1) Project Creator:

Project Creator component let user create and edit his/her own project.

1) Project Creation: This service is about registered user creating new donation projects. It is provided by Project Creator subsystem component and required by User Operations subsystem component.

2) Project Editing: This service is about editing existing project created by the User. This service provided by Project Creator subsystem component. and required by User Operations subsystem component.

Both of these services provided by Project Management Subsystem and required by User Management Subsystem.

2) Administrative Project Operator:

This subsystem component is about verifying project created by Users. This subsystem component has relevance with the Admin since Admin is responsible for verifying recently created project.

1) Project Verification: This service verifies recently created donation projects by User. The service is provided by Administrative Project Operator subsystem component and required by Project Creator subsystem component. This service use within the Project Management Subsystem.

Donation Management Subsystem:

This subsystem contains two components. One is Donation Creator and other is Donation Tracker. Donation Creator component provide one service called Start Ongoing Donation. Donation Tracker component provide two services called Track Cargo and Complete Donation.

1) Donation Creator:

This component let user start his/her own donation.

1) Start Ongoing Donation: This service starts a new donation process. Donation process initiated by User. When donation process begun by the User, Donation creator component provides Start Ongoing Donation Service and the service is acknowledged by User Operations component.

2) Donation Tracker:

This component able User either track his/her donation process or his/her finish the donation process. Here, two services provided by this component will be discussed.

1) Complete Donation: This service completes the donation process already initiated by the User. User can track his/her donation and begin to donation completion process. Provided by

Donation Tracker component and required by User Operations component. When User complete donation, donation tracker initiates Complete
Donation service. User Operations component require Complete Donation.

2) Track Cargo: This service able User tracking or view his/her cargo send as donation. When User wants to track his/her cargo, Donation Tracker component provides Track Cargo service to Profile Creator. Profile Creator component requires Track Cargo Service.

Storage Management Subsystem:

Storage Management Subsystem provide data management services to system. Every information or data about the system are related with this Storage Management Subsystem. This subsystem has three different components. User Storage, Donation Storage and Project Storage. In addition, this subsystem's services are not only store data also these services able user of the WWDS system can gather data from Storage Management Subsystem.

1) Donation Storage:

This component manages data about each User's ongoing donation.

1) Store Ongoing Donation: Store data about ongoing donation information. This service makes system store information about recently started donation. Donation Storage provide Store Ongoing Donation service and the service is required by Donation Creator component.

2) User Storage:

User storage stores information about User. We can gather information from User Storage as well.

1) User Recording: When User registers, system store information. User Recording service provided by User Storage component and required by Register component.

2) User Verification: When User login function, system check login information given by User. If system matches information given by User and information in user database matches then User login to the system. User Verification service provided User Storage component and required by Login component.

<WWDS>

3) Project Storage:

Project Storage component manage data about each individual project.

1) Store Project: When project creation is done and verified, information about project is stored in the database. Provided by Project Storage component and required by Project Creator.

2) Store Cargo: Manages data about cargo send by User. Cargo can be stored in database or User who received the cargo, can gather information from database. Provided by Project Storage component and required by Donation Tracker component.

3) Project Search: When User wants to search information about project, WWDS gathers information from WWDS database then show this information to User. Project Storage provide this service and User Operations component require this service.

5.References

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

2. http://whatis.techtarget.com/glossary/Programming

3. Programming Visual Basic .NET Jesse Liberty 2nd Edition.