<Survey Horse>

System Design

<1.0>

<09.12.2016>

Osman Çiçek

Emre Şaşmaz

Burak Sağlam

Ali Buğra Kanburoğlu

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc433996772)

[1.1. Purpose of the System 1](#_Toc433996773)

[1.2. Design Goals 1](#_Toc433996774)

[1.3. Definitions, Acronyms, and Abbreviations 1](#_Toc433996775)-2

[1.4. References 1](#_Toc433996776)

[2. Current Software Architecture 2](#_Toc433996777)

[3. Proposed Software Architecture 3](#_Toc433996778)

[3.1. Overview 3](#_Toc433996779)

[3.2. System Decomposition 4](#_Toc433996780)

[3.3. Hardware Software Mapping 4](#_Toc433996781)

[3.4. Persistent Data Management 4](#_Toc433996782)

[3.5. Access Control and Security 5](#_Toc433996783)

[3.6. Global Software Control 5](#_Toc433996784)

[3.7. Boundary Conditions 5-6](#_Toc433996785)

[4. Subsystem Services 6](#_Toc433996786)

[5. References 6](#_Toc433996787)

SYSTEM DESIGN DOCUMENT [1]

# Introduction

## Purpose of the System

In this system, we designed a survey system to provide users that want to make survey and with the powerful, conscious tools that we use, They will make meaningful statistical analysis according to complexity and will be able to analyze them with well-designed graphics. Our purpose is to design a system that users in the all age group can use so our design based on simplicity, abstraction and effectivity. In addition to this, we have designed a model that also aimed to please both researchers and analysts to provide fast, reliable, lossless and meaningful data. By developing this system we tried to serve any type of survey, any topic to make our system preferable by anyone. We designed the system for researchers and analysts to reduce time wastage. We create real-time result panel to decrease wasting time specifically for the researchers and analysts. So they can see the analysis results like professional researchers.

## Design Goals

Since all devices able to connect internet so we have implemented our system as web-based. It is very easy to use; define a questions, create a survey, share it and wait your survey to be filled by visitors. Our system has there mainly parts. First of them for the researchers, second one for the users with registration and last one for the users who only fills the survey without registration. Survey Horse can use everyone because there is no payment. Our system has a good purpose that is completely free and user-friendly. Our domain includes https and security certificate which are provide secure and reliable information. We have also SSL (Secure Sockets Layer) and some other security and back-up services to protect your data.

## Definitions, Acronyms, and Abbreviations

Important terms and concepts are listed here.

*Distribution*

There are answers to survey`s questions so options have some rate to use during

decision making.

*Domain*

Address and name ofwebsite.

*GUI*

Graphical User Interface.

*HTTP*

It is a protocol for secure communication over a computer network which is widely used on the Internet.

*Model*

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

*MVC*

Model/View/Controller architectural style.

*Researcher*

If user want to research about anything, they can create survey in the system.

*Subsystem*

A part of decomposed system.

*Survey*

Survey created by the surveyor.

*Surveyor*

Each user can have a relationship with the system.

*System*

Any interactions performed by the application are considered to be performed by

the system.

*Undo*

If the user fill in the blank or answer the questions wrongly, they can reverse that action.

*View*

A visual representation of a model which might*.*

## References

* Requirements Analysis Document (20.11.2016)
* Surveymonkey.com

# Current Software Architecture

Describe the architecture of the system being replaced**. If there is no previous system**, this section can be replaced by **a survey of current architectures for similar systems**. The purpose of this section is to make explicit the background information that system architects used, their assumptions, and common issues the new system will address.

# Proposed Software Architecture

Survey horse is a questionnaire system that used for statistical analysis and to have an opinion on any matter. They can created survey about research topics only with a computer connected the network.

The system serves for the use of system admin, survey manager, registered surveyor, unregistered surveyor with respectively changing boundaries for each user. Every user except unregistered surveyor in the system has some common prerogatives such as being register, login and also, they have some specific prerogatives too. Thanks to the system`s authority gives to all users type edit their name, username, password, birthdate and email whenever they need.

The system also gives authority system admin and Survey Manager to show results of survey. System Admin can reach all surveys and can run them. Actually, system admin has a few the more privileges than system survey manager because, they might also manage the all users. The other user type is unregistered surveyor in the system which have a role that fill in survey.

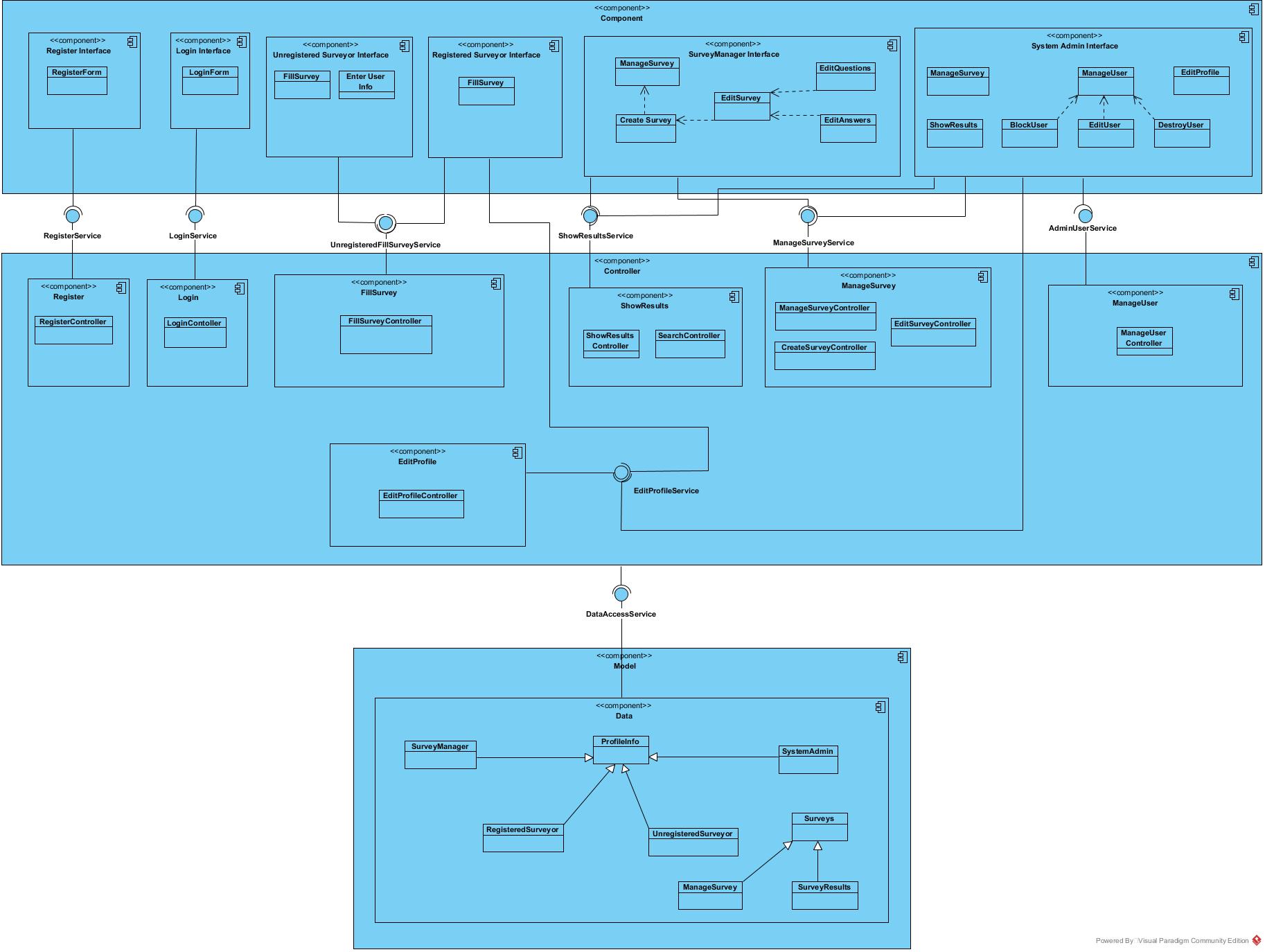
## Overview

Our Survey HORSE system will be developed according to the Model/View/Controller architectural style. 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.

## System Decomposition



## Hardware Software Mapping

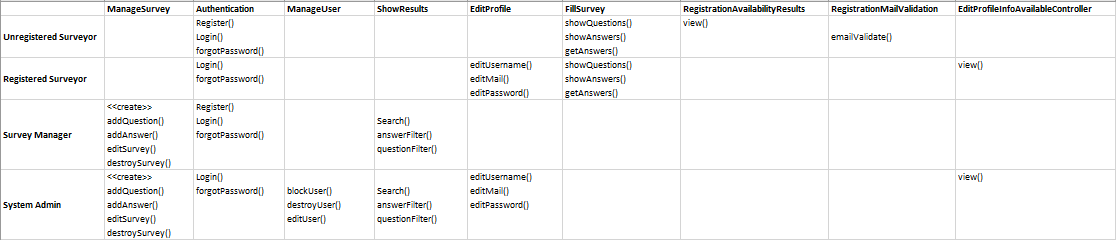
Survey Horse is built as a web based system. Hence, the system must be deployed on the server. The system which is deployed on the same server as the database will communicate to each other when it requires to access to the database. The server will be configured to reply the database-related requests from its default MySQL.

## Persistent Data Management

We need to store user information; name, surname, email, password of every single user. In addition, we hold the all surveys and all answers so we have decided to use MySQL to create relational database.

Ruby supports both SQLite and MySQL but SQLite works in local and may cause vulnerability so we won’t use it. We try to design a well-structured database such as 3NF or BCNF. Our queries will work faster and will not have any collusion. So it will affect the system efficiency. Client will have faster results and faster surveys.

## Access Control and Security



**Authentication:**

While become to login in the survey horse system check if username and password are valid, the system will call the “Login()” function. If the values are same with the database, the system will allow to login, if they do not same then the system will show an error message.

**Authorization:**

In our system, we provided authorization to some actors in order to do their responsibilities. It protects the actor to reach the other actors’ job. For example, unregistered Surveyor cannot use the “Manage User” subsystem because the authorization is given to only “System Admin”. So, every actor has their own authorization. We keep them into database separately.

## Global Software Control

Survey Horse is a web application that use ruby on rails so our system is based object-oriented approach.

In Survey Horse, is very easy to use all users because the system has a little operations also there is no need any installing one of the most important thing is register to create survey.

There is need to define admin that is seed after the implement code. The system admin should manage all of web sites. The second event register users and survey manager should be register in the system if they don’t login to manage, create or answer survey they can’t these operations.

According to the system is very simple to use because there is nothing to use. After code implement had done the system will be ready to operate.

## Boundary Conditions

When the system works, system admin logins automatically and first users of this system; unregistered surveyor, survey manager can register the system. If Unregistered Surveyor registered to the system, then she/he will become a registered surveyor. After they login, they can call every functions which are allowed to be used.

We have some control objects such as entering wrong username/password in login and register page or entering invalid e-mail, username, password in register page. We have applied the same control objects to the “Edit Information” page. We also check the survey’s questions to make sure questions are correct (spell check) and not include any abuse word/sentence. In addition, we check the answer options whether they are all the same or they are fake.

Admin can shut down the system and when it is off, system cannot be used by anyone including system admin. After this point, System Admin only can start the system to make it in use again.

# Subsystem Services

We have used Model View Controller (MVC) Architecture style. In system decomposition part, we have divided the system into several subsystems. It provides us to see which subsystem gives service/services to the other and which one gets service/services from the other. This is significant to understand how system works. Also thanks to this, we can observe our system from bird’s eye view and see the wrong/missed design if any.

For Example in our subsystem design, “Register Subsystem” has a form in view layer and has a controller in controller layer and finally in the model layer, we keep the register information here. When we check the “Survey Manager” subsystem, we can see its methods inside the component. Cause both “System Admin” and “Survey Manager” can manage the surveys, “Manage Survey Controller” gets service from both “System Admin Interface” and “Survey Manager Interface”. On the other hand, because “Manage User Controller” only reachable by the System Admin, it gets service from only “System Admin Interface”. In addition, if we check the “Show Results” controller, because it is reachable from both “System Admin” and “Survey Manager”, it gets services from both “Survey Manager Interface” and “System Admin Interface” In order to fill the survey, there will be a single controller which is “Fill Survey Controller” gets service by “Unregistered Surveyor Interface” and “Registered Surveyor Interface”. Finally, “Edit Profile Controller” can be usable by “Registered Surveyor” and “System Admin”, it gets services from “System Admin Interface” and “Registered Surveyor Interface”. In the Model layer, we keep some information like “Profile Info”, for the information of all users, “Survey Results” for the results of surveys, “Survey” for the all surveys. Both “Manage Survey” and “Survey Results” are inherits the “Survey”. We also keep all actors information separately which are interface of “Profile Info”.

# 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.
2. https://www.surveymonkey.com/