`

DESIGN REPORT ON SOFTWARE MAINTAINABILITY

GuessWho!

Face Recognition Game to aid Dementia Patients

Delivered by
Bookies
Joshi Chaitanya Krishna(U1522971F)
Genevieve Lam Wen Qi (U1521863H)
Sharma Vidur (U1522940D)
Yong Chen Feng (U1620913B)
Heng Zhi Guang (U1620660F)
Chen Guanyu (U1621093D)

Submitted to –
Dr Shen Zhiqi
Nanyang Technological University
School of Computer Science and Engineering

`

Revision History

Version Number	Date	Author(s)	Comments
1.0	21 March 2017	Team Bookies	

`

Table of Contents

Revision History 3

1 Introduction 5

2 Architectural Pattern 5

2.1 Model 5

2.2 View 5

2.3 Controller 6

3 Design Patterns 6

3.1 Client and Server Pattern 6

3.2 Observer Pattern 7

4 System Requirement Specification 7

5 Software Configuration Management (SCM) 7

6 Software Quality Attributes 8

`

1 Introduction

Software maintainability is defined to which application is repaired or enhanced. The purpose is to ensure that our code is portable across different system configurations as well as the ease of implementing new features without making drastic changes.

With high maintainability, faults in our application are easily diagnosed and patching an update to fix the related issue can be done in a fast and efficient manner.

Software maintainability is determined by how easy the developer make changes to the program without losing its main functionalities.

Software maintainability is achieved by adopting Software Design Principles like Open Close Principle, low coupling and high cohesion.

2 Architectural Pattern

Our application is structured according to the Model-View-Controller (MVC) architecture. It divides the application into three interconnected parts for separating internal representation of information. By adopting the MVC architecture, modular code can be achieved.

2.1 Model

Model represents the shape of the data and business logic. It maintains the data of the application and its object, retrieve and store model state in the database.

2.2 View

A View is a user interface. View display data using model to the user and it also enables them to modify the data. Such representing elements in user interfaces includes buttons and display boxes.

In web based MVC systems, a view can be implemented using a template that renders an HTML page. In our face matching game application, the Views would be the templates for rendering the login page, the matching page etc.

In our application, the following views are used in the code:

- caretaker-dashboard.ejs
- caregiver-image.ejs
- caregiver-menu.ejs
- caregiver-upload.ejs
- correctguess.ejs
- index.ejs
- login.ejs
- patientmenu.ejs
- register-caregiver.ejs
- scores.ejs
- signup.ejs
- wrongguess.ejs

`

2.3 Controller

Controller handles the user request and translates them into actions that the Model should take. It renders the appropriate view with the model data as a response.

In a web environment, this is usually done by having the Controller to handle incoming HTTP requests. It is possible to have more than one Controller in the application.

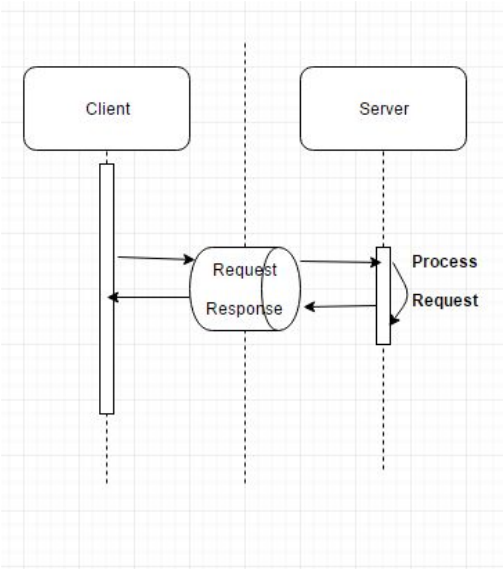
The following controllers are used in the code:

- controller.js
- populateDB.ejs
- populateImages.js
- routes.js

3 Design Patterns

3.1 Client and Server Pattern

GuessWho! adopts client and server design pattern, especially the request and response subtype patterns. The figure below depicts the process:



When the application need to perform an action that requires the service, it sends a request to the server. The server will then perform the action and transmit a response back to the application.

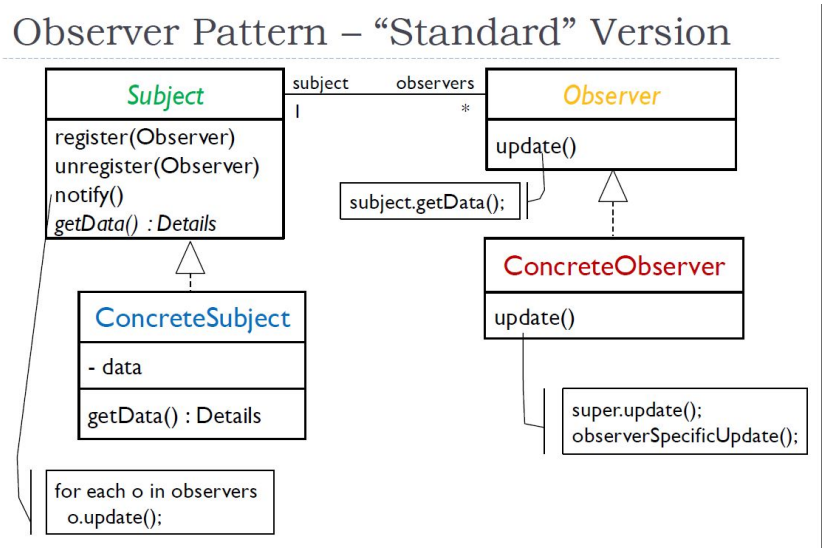
In addition, when the application sends a request to the database, it will process the request in the backend and generate a response back to the application.

Since the request is sent to the server asynchronously, the client and server pattern provides a fast and simple process.

`

3.2 Observer Pattern

Observer Pattern is one of many software design patterns in which an object called the subject maintains a list of dependents called observers. It notifies them automatically if there is a state change by calling its method.



This pattern is implemented in our application. Users have to match the face show on the display and click the correct naming for it. Whenever a button is clicked, the corresponding listener of the button is invoked. Observer pattern helps in decoupling different objects that interact with each other and therefore, promoting maintainability.

This pattern is also implemented internally in router. Whenever a routing path is called, the route call back function corresponding to the routing path is called. The advantage of adopting this pattern is the decoupling of different request that interact with each other, thus promoting maintainability. This is achieved using encapsulation by properly organizing the code in the correct places

4 System Requirement Specification

System Requirement Specification (SRS) document contains the design and implementation such as class diagram and activity diagram. It serves as a reference document to educate developers who maintain GuessWho! and facilitate maintenance activities that developers who have limited understanding of the application.

5 Software Configuration Management (SCM)

SCM objective is to help teams to maintain consistency and control over what is produced. It ensures that correct product versions are delivered, satisfy the requirements and supporting materials such as hardware and documents are consistent. SCM also practices revision control. For example, if something goes wrong, SCM can determine what was changed and who done the changes. Maintainers are required to use Configuration Management Plan as a guide to conduct any SCM processors.

`

6 Software Quality Attributes

There are many software quality attributes taken into consideration when designing on the application. The table bellows shows some of the quality attribute that relates to design qualities.

Category	Quality attributes	Description
Design Qualities	Conceptual integrity	Consistency and coherence of the program. For example, how components or module are designed, coding style and variable naming.
	Maintainability	Ability of the system to undergo changes with a degree of ease. When changes are made, components, service feature as well as interfaces might be affected causing it malfunction. Therefore, application must have documentation or manual so that in future new developer wants to make changes to the system and they are not familiar with the coding they might take a huge amount of time learning about the system.
	Reusability	Defines capability for components suitable for use in other applications and in other scenarios. This minimizes duplication of components and implementation time.