**UB (pseudonym)**

**Software Requirements Specification**

**and**

**Project Proposal**

**Since June 2015**

**Last Edit 08.03**

**Version 1.3.1**

**Engine**

**강지현**

본 문서에 가칭 UB 프로그램 개발에 관하여 명세한다.

본 문서의 일부 또는 전체에 대해 무단 복제를 금한다.

2015년 여름방학(7월-8월) 중 본 프로젝트를 1차 완성하는 것을 목표로 한다.

INDEX

[1. Introduction 5](#_Toc426486960)

[1.1 Purpose 5](#_Toc426486961)

[1.2 Scope 5](#_Toc426486962)

[언어 변화의 기록장 5](#_Toc426486963)

[민주적 언어 결정 5](#_Toc426486964)

[빠른 언어변화 대응 5](#_Toc426486965)

[1.3 Glossary 5](#_Toc426486966)

[1.4 References 5](#_Toc426486967)

[2. Overall description 6](#_Toc426486968)

[2.x Product Perspective 6](#_Toc426486969)

[2.x System Environment 6](#_Toc426486970)

[2.x Architectural Design 6](#_Toc426486971)

[2.x.x MVC Framework 6](#_Toc426486972)

[2.x.x Application Detached from System 6](#_Toc426486973)

[2.x Rules and Constraints 6](#_Toc426486974)

[2.x.x Naming Rules 6](#_Toc426486975)

[2.x.x Convention 6](#_Toc426486976)

[2.x URL Routing Outline 6](#_Toc426486977)

[2.x.x URL Contract 6](#_Toc426486978)

[2.x Database Design 7](#_Toc426486979)

[2.x.x Table Diagram 7](#_Toc426486980)

[2.x.x Table Description 7](#_Toc426486981)

[Term (Table) 7](#_Toc426486982)

[Member (Table) 7](#_Toc426486983)

[Message (Table) 7](#_Toc426486984)

[3. System Features 7](#_Toc426486985)

[3.1 Object and Flow Diagram 7](#_Toc426486986)

[3.2 System Components Overview 8](#_Toc426486987)

[3.2.1 Gateway and Configuration 8](#_Toc426486988)

[Gateway (root/index.php) 8](#_Toc426486989)

[Config.xml 8](#_Toc426486990)

[3.2.2 Core 8](#_Toc426486991)

[Core.php 8](#_Toc426486992)

[Router.php 9](#_Toc426486993)

[Controller.php 9](#_Toc426486994)

[Model.php 9](#_Toc426486995)

[Database.php 10](#_Toc426486996)

[Data.php (ABORT) 10](#_Toc426486997)

[3.2.3 Controller 10](#_Toc426486998)

[Root.php 10](#_Toc426486999)

[Term.php 10](#_Toc426487000)

[About.php 11](#_Toc426487001)

[Rules.php 11](#_Toc426487002)

[Guide.php 11](#_Toc426487003)

[Notice.php 11](#_Toc426487004)

[Archive.php 12](#_Toc426487005)

[Login.php 12](#_Toc426487006)

[3.2.4 Model 12](#_Toc426487007)

[Term\_md.php 12](#_Toc426487008)

[Comment\_md.php 13](#_Toc426487009)

[SNSBrowser\_md.php 13](#_Toc426487010)

[Notice\_md.php 13](#_Toc426487011)

[Archive\_md.php 14](#_Toc426487012)

[Member\_md.php 14](#_Toc426487013)

[Document\_md.php 14](#_Toc426487014)

[3.2.5 View 14](#_Toc426487015)

[3.3 Performance Requirements 14](#_Toc426487016)

[3.4 Other Requirements 14](#_Toc426487017)

# 1. Introduction

## 1.1 Purpose

본 문서는 UB (가칭)의 개략적, 그리고 상세한 서술을 위해 작성되었다. 왜 프로젝트가 기획되었는지, 어떤 방식으로, 어떤 기능을 갖출 것인지 설명한다.

## 1.2 Scope

2015년 상반기 현재 국내에 있는 웹 서비스 중, 한국어의 사용의, 더 자세히 말하여, 신조어와 단어의 변화에 대한 경과를 기록하고 토의하는 Platform이 없어, 이와 같은 기능을 할 수 있는 웹 서비스를 개발하기로 하였다.

UB는 신조어와 파생어의 단어를 사용자로부터 제공받는 웹 어플리케이션이다. 사용자는 자신의 의지에 따라, 단어의 의미를 제공할 수 있고 또한 타인이 제시한 의미를 열람할 수 있다. 같은 단어의 의미는 점수 알고리즘에 의거, 점수의 오름차순으로 보여진다.

#### 언어 변화의 기록장

새로이 생성되는 표현 혹은 단어와 그 의미를 기록함에 여러 의미가 있다. 첫째, 언어의 변화 양상을 추적할 수 있다. 언어는 끊임없이 변화한다. 본 서비스는 변화하는 언어에 대한 지식을 언어 사용자 스스로 제공케 하여, 이용자 전체로 하여금 빠르고 효율적인 방법으로 변화 요소를 습득, 인지하게 한다. 또한 이 기록은 훗날 말뭉치로서 언어와 문화연구 등 다방면에 기여할 것이다.

#### 민주적 언어 결정

UB는 특정 권위체에 의해서가 아닌 이용자 불특정 다수로부터, 투표에 의해 의미를 형성하는 방식을 채택한다. 이는 민주적인 방식의 언어 형성을 돕는다. 다시 말해, 표현과 단어들은 공동의 합의에 의해 선택된다.

#### 빠른 언어변화 대응

현재 World Wide Web 과 무선인터넷의 보급이 과거 어느 시점보다 넓게 돼 있다. 디지털 소통 채널은 개인의 언어 사용빈도를 증가시켰고, 또한 채널의 특수성-구두가 아닌 Text를 통한 소통은 다양한 형태의 언어 Variation을 야기하고 있다. 변화가 전보다 매우 빠르다. 이에 대응하여, 전의 “사전”을 이용하는 것에 효용이 있지 못함을 인지할 수 있다. UB는 훌륭한 대안이 될 수 있다.

새로운 사전

잊혀질 권리

## 1.3 Glossary

본 프로젝트 명세에 등장하는 용어와 그 설명을 밝힌다.

|  |  |
| --- | --- |
| **Word** | **Description** |
| localhost | 도메인 혹은 root가 되는 주소이다. |
| root | 웹 서버 최상위 폴더 |
| UB | 본 프로젝트의 가명, 웹 사이트인 동시에 웹 프로그램을 지칭하는 용어이다. |
| singleton | Singleton Form. 객체를 단 한 개만 생성하는 형태의 프로그래밍 기법이다. |
|  |  |

## 1.4 References

추후 서술

# 2. Overall description

## 2.x Product Perspective

## 2.x System Environment

php

mysql

html5

css3

javascript

jquery

ajax

후에 상세서술

## 2.x Architectural Design

### 2.x.x MVC Framework

UB는Model View Controller Framework에 입각해 설계한다. Model은 데이터 가공을, View는 화면 출력을, 그리고 Controller는 Model과 VIew를 포함한 전체 Logic을 관장한다.

### 2.x.x Application Detached from System

System과 Application을 분리해서 설계하고, System에 필수적인 기능만 구현한 후, Application이 이를 상속하여 실행될 수 있도록 한다. 이렇게 디자인하는 가장 큰 이유는 요소별 모듈화를 통해 이식을 최대한 효율적으로 하기 위함이다.

## 2.x Rules and Constraints

### 2.x.x Naming Rules

파일은 단일객체만을 가지고 있거나, 아무런 객체도 가지고 있지 않은 것으로 나뉜다. 객체를 가지고 있는 경우, 객체의 이름과 파일 명은 모두 첫 글자를 대문자로 시작한다. Controller의 경우 객체명을 위와 같은 규칙에 의거해 그대로 쓰고, 모델의 경우 말미에 \_md를 붙인다. View프로그램의 경우 첫 글자가 소문자이다.

### 2.x.x Convention

## 2.x URL Routing Outline

### 2.x.x URL Contract

URL은 domain/segment1/segment2/segment3,,.형식이다. UB는 요청 받은 URL을 토대로 Routing을 통해 적절한 Contents를 제공한다. URL은 다음과 같이 요소화한다.

segment[0]/segment[1] /segment[2]/,,.

= **domain / CONTROLLER / PARAMETER1 / PARAMETER2,,.**

즉, segment[1] 에 요청될 컨트롤러가 위치하고, 그 이후 Parameter들로 취급한다.

이와 같은 URL은 사용자 직관성을 우선적으로 고려하여 설정하였으며, MVC Framework 기반의 개발과도 맥락을 같이 한다.

## 2.x Database Design

### 2.x.x Table Diagram

### 2.x.x Table Description

#### Term (Table)

Term은 단어와 의미 정보를 가지고 있는 테이블이다.

Term Table은 다음의 항목들로 구성되어 있다.

id: 단어의 고유 index

lemma: lemma. 단어가 지칭하는 추상적인 개념

word: 단어의 표면적 형태

def: 의미

date: 올린 날짜

like: 좋아요를 받은 수

dislike: 싫어요를 받은 수

hit: 조회 수

pop: like, dislike, hit을 토대로 UB의 고유 알고리즘에 의해 계산한 인기도 지수.

#### Member (Table)

Member은 회원 정보를 가지고 있는 테이블이다.

Member Table은 다음의 항목들로 구성되어 있다.

id: 회원 고유 index.

email: 이메일주소. 곧 회원의 접속 아이디.

pw: 패스워드

nickname: 닉네임

intro: 소개

photo: 사진 혹은 그림

registerDate: 가입일

lastLogin: 최종 로그인

like: 이제까지 좋아한 것.

dislike: 이제까지 싫어한 것.

rating: 알고리즘으로 계산한 평판.

#### Message (Table)

Message은 사용자간 메시지 정보를 가지고 있는 테이블이다.

Message Table은 다음의 항목들로 구성되어 있다.

추후 서술

# 3. System Features

## 3.1 Object and Flow Diagram

*다음 장에 전체 Object Diagram과 Flow Chart를 첨부한다.*

## 3.2 System Components Overview

본 프로그램은 다양한 구성 요소로 이루어진다. 주요 성분으로 Gateway, Configuration, Core, Controller, Model과 View등이 있다. 이하 이에 대한 내용을 기술한다.

### 3.2.1 Gateway and Configuration

#### Gateway (root/index.php)

localhost 가 호출되면 가장 먼저 root 폴더의 index.php 를 실행한다. 즉, root/index.php 가 UB의 Gateway 기능을 한다. 또한, 새로운 URL 로의 이동이 있을 시 (예: Anchor을 통한), 다시 가장먼저 Gateway를 실행한 후 요청에 맞게 프로그램을 실행한다. 즉, Script를 이용한 페이지 요소 갱신 제외, 모든 링크 이동은 반드시 Gateway를 요청한다.

|  |  |
| --- | --- |
| ***Procedure*** |  |
| *Path Setting* | 시스템 전반에 필요한 Path를 설정한다. |
| *Loading Files* | debug.php, define.php, core.php 로드한다. |
| *Initiate Core* |  |
|  |  |

#### Config.xml

Config.xml은 UB 실행에 관한 여러 Configuration 정보를 갖고 있는 XML형식의 파일이다.

Config.xml이 가지고 있는 XML Element에 관한 명세는 다음과 같다.

|  |  |
| --- | --- |
| ***Components*** |  |
| */Info* | 시스템의 전반적 정보를 담는다.  하위 속성: name, version, root. |
| */Database* | Database 객체가 필요한 정보들로 데이타베이스와 관련한 정보를 가지고 있다.  하위 속성: name, host, port, user, password |
| */Router* | Router 객체가 필요한 정보들.  하위 속성: url, controller, method |
|  |  |
|  |  |

### 3.2.2 Core

#### Core.php

Core는 UB의 최상위 Managing Object 이다. Core는 Config객체와 Router객체를 갖고, Router객체의 route()를 실행하는 기능을 한다.

|  |  |
| --- | --- |
| ***Field*** |  |
| *$instance []* |  |
| *$config* |  |
|  |  |
| ***Method*** |  |
| *\_\_construct($file\_config)* |  |
| *spl\_autoload\_register(function($class){})* |  |
| *getInstance()* |  |
| *getControllerParam()* |  |
| *main()* |  |
|  |  |

#### Router.php

URL 정보를 파싱하는 중추적인 역할을 하는 객체이다. 해당 기능을 하는 route()를 포함한다.

|  |  |
| --- | --- |
| ***Field*** |  |
| *$\_routes* |  |
|  |  |
|  |  |
| ***Method*** |  |
| *\_\_construct($config\_Router)* |  |
| *toRoutes($url, $ctrl, $method)* |  |
| *match($\_routes, $url, $method)* |  |
| *route()* |  |
|  |  |
|  |  |

#### Controller.php

MVC의 C에 해당하는 요소로, Model과 View를 포함하고 전체적인 로직을 관장하는 객체이다. 모든 Controller는 main 함수를 기본함수로 가진다. 사용자 정의 Controller 객체들은 모두 이 Controller 객체를 상속한다.

|  |  |
| --- | --- |
| ***Field*** |  |
| *$view* |  |
| *$model []* |  |
| *$data []* |  |
| *$\_CONSTANT []* |  |
|  |  |
| ***Method*** |  |
| *\_\_construct()* |  |
| *main()* |  |
| *\_\_set($name, $value)* |  |
| *\_\_get($name)* |  |
| *redirect($url, $permanent)* |  |
| *render($file\_template)* |  |

#### Model.php

MVC의 M에 해당하는 요소로, 데이터 입출력을 관장한다. 모든 사용자 정의 Model 객체들은 이 객체를 상속한다.

한 개의 모델은 한 개의 DB테이블과 교신하는 것을 원칙으로 한다.

|  |  |
| --- | --- |
| ***Field*** |  |
| *$db* |  |
|  |  |
| ***Method*** |  |
| *getDatabase()* |  |

#### Database.php

Model이 소유하고 있는 Database객체로, DB와 접속을 관장하는 객체이다.

|  |  |
| --- | --- |
| ***Field*** |  |
| *$name* |  |
| *$host* |  |
| *$port* |  |
| *$user* |  |
| *$password* |  |
|  |  |
| ***Method*** |  |
| *\_\_construct($config\_Database)* |  |
| *getConnection()* |  |
| *\_\_clone()* |  |

#### Data.php (ABORT)

Model이 소유하고 있는 Data객체로, DB에서 가져온 데이터를 가지고 있다.

Data.php는 다음의 요소를 포함한다.

함수

set():

get():

### 3.2.3 Controller

#### Root.php

Domain이 호출되면 기본값으로 출력되는 화면을 구성하는 컨트롤러 객체다.

|  |  |
| --- | --- |
| ***URL*** | localhost/ |
|  |  |
| ***Field*** |  |
| *$model []* |  |
| *$view* |  |
| ***Method*** |  |
|  |  |
|  |  |

#### Term.php

URL: localhost/term/

검색 요청을 관장하는 컨트롤러로, 사용자가 요청하는 Term을 추출하여 보여주는 기능을 한다.

|  |  |
| --- | --- |
| ***URL*** | localhost/term/\* |
|  |  |
| ***Field*** |  |
| *$model []* |  |
| *$view* |  |
| ***Method*** |  |
|  |  |
|  |  |

#### About.php

URL: localhost/about/

UB에 관한 개략적 정보를 출력하는 것을 관장하는 컨트롤러이다.

|  |  |
| --- | --- |
| ***URL*** | localhost/about/\* |
|  |  |
| ***Field*** |  |
| *$model []* |  |
| *$view* |  |
| ***Method*** |  |
|  |  |
|  |  |

#### Rules.php

URL: localhost/rules/

UB의 이용 규칙 전반을 출력하는 것을 관장하는 컨트롤러이다.

|  |  |
| --- | --- |
| ***URL*** | localhost/rules/\* |
|  |  |
| ***Field*** |  |
| *$model []* |  |
| *$view* |  |
| ***Method*** |  |
|  |  |
|  |  |

#### Guide.php

URL: localhost/guide/

UB를 어떻게 이용해야 하는지 소개하는 것을 관장하는 컨트롤러이다.

|  |  |
| --- | --- |
| ***URL*** | localhost/guide/\* |
|  |  |
| ***Field*** |  |
| *$model []* |  |
| *$view* |  |
| ***Method*** |  |
|  |  |
|  |  |

#### Notice.php

URL: localhost/notice/

최근 공지사항을 출력하는 것을 관장하는 컨트롤러이다.

|  |  |
| --- | --- |
| ***URL*** | localhost/notice/\* |
|  |  |
| ***Field*** |  |
| *$model []* |  |
| *$view* |  |
| ***Method*** |  |
|  |  |
|  |  |

#### Archive.php

URL: localhost/archive/

계층 별, 또는 다른 군 별 단어 검색 분포와, 최근 신조어 동향 등 큰 단위의 Data를 수집하고, 분석하고, 출력하는 것을 관장하는 컨트롤러이다.

|  |  |
| --- | --- |
| ***URL*** | localhost/archive/\* |
|  |  |
| ***Field*** |  |
| *$model []* |  |
| *$view* |  |
| ***Method*** |  |
|  |  |
|  |  |

#### Login.php

URL: localhost/login/

사용자가 UB에 로그인 혹은 회원가입을 요청 시, 이를 관장하는 컨트롤러이다.

|  |  |
| --- | --- |
| ***URL*** | localhost/login/\* |
|  |  |
| ***Field*** |  |
| *$model []* |  |
| *$view* |  |
| ***Method*** |  |
|  |  |
|  |  |

### 3.2.4 Model

#### Term\_md.php

단어 검색 요청 시, UB의 DB로 부터 해당 요청을 처리하는 모델이다.

|  |  |
| --- | --- |
| ***Field*** |  |
| *$data []* |  |
|  |  |
| ***Method*** |  |
| *getTerm()* |  |
| *getTermExact()* |  |
| *getRecentTerm()* |  |
| *getSuggestedTerm()* |  |
| *addTerm()* |  |
| *getOtherDef()* |  |
| *getMyTerm()* |  |

#### Comment\_md.php

단어 검색 요청 시, UB의 DB로 부터 해당 요청을 처리하는 모델이다.

|  |  |
| --- | --- |
| ***Field*** |  |
| *$data []* |  |
|  |  |
| ***Method*** |  |
| *getComment()* |  |
| *getMyComment()* |  |
| *getCommented()* |  |
| *addComment()* |  |
| *delComment()* |  |
| *editComment()* |  |

#### SNSBrowser\_md.php

특정 커뮤니티, 포털 사이트들로부터 Term에 관한 용례를 가져오는 기능을 관장하는 모델이다.

추후 설명

#### Notice\_md.php

공지사항에 있는 자료를 처리하는 모델이다.

|  |  |
| --- | --- |
| ***Field*** |  |
| *$data []* |  |
|  |  |
| ***Method*** |  |
| *getNotice()* |  |
| *getNoticeExact()* |  |
| *addNotice ()* |  |
| *getOtherDef()etMyTerm()* |  |
| *addTerm()* |  |
|  |  |

Notice\_md.php 는 다음의 요소를 포함한다.

함수

getNotice(): 공지사항 목록을 가져온다.

getNoticeExact(): 특정 공지사항을 가져온다.

delNotice(): 공지사항 항목을 삭제한다.

addNotice(): 공지사항 항목을 작성한다.

editNotice(): 공지사항 항목을 수정한다.

#### Archive\_md.php

추후 서술

|  |  |
| --- | --- |
| ***Field*** |  |
| *$data []* |  |
|  |  |
| ***Method*** |  |
|  |  |
|  |  |

#### Member\_md.php

회원정보를 관장하는 모델이다.

|  |  |
| --- | --- |
| ***Field*** |  |
| *$data []* |  |
|  |  |
| ***Method*** |  |
|  |  |

#### Document\_md.php

추후 서술

|  |  |
| --- | --- |
| ***Field*** |  |
| *$data []* |  |
|  |  |
| ***Method*** |  |
|  |  |
|  |  |

### 3.2.5 View

추후 서술

## 3.3 Other Requirements

추후 서술