public function cleanInput(\$input) public function validateInput(\$input, \$fieldName, \$maxLength, \$nullAllowed)

class DepartmentBusinessLogic private \$departmentDao = null; private \$businessLogic = null; private \$maxDepCodeLenth = 10; private \$maxDepNameLength = 100; public function __construct() public function getAllDepartments() public function getAllDepartmentsAlphabetical() public function getDepartmentById(\$departmentId) public function getDepartmentByCode(\$departmentCode) public function getAssocAdmins(\$departmentId) public function getAssocQuestions(\$departmentId) public function getAssocPrograms(\$departmentId) public function addDepartment(Department \$department) public function updateDepartment(Department \$department)

public function deleteDepartment(Department \$department)

public function cleanDepartment(Department \$department)

public function validateDepartment(Department \$departme class AdminBusinessLogic private \$adminDao = null; private \$businessLogic = null; private \$maxNameLength = 40; private \$maxPasswordLength = 100; public function __construct() public function getAllAdmins() public function getAdminById(\$adminId) public function getAdminByUsername(\$username) public function getAssocDepartments(\$adminId) public function addAdmin(Admin \$admin, \$departments) public function updateAdmin(Admin \$admin, \$departments)

public function deleteAdmin(Admin \$admin)

public function cleanAdmin(Admin \$admin) class QuestionBusinessLogic private \$questionDao = null; private \$businessLogic = null; private \$maxNameLength = 200; public function __construct() public function getAllQuestions() public function getQuestionById(\$questionId) public function getQuestionByText(\$questionText) public function getAssocDepartments(\$questionId) public function getAssocAnswers(\$questionId) public function addQuestion(Question \$question, \$departments public function updateQuestion(Question \$question, \$departments public function updatePosition(\$departmentId, \$q1, \$q2) public function deleteQuestion (Question \$question) public function cleanQuestion(Question \$question)

public function validateQuestion(Question \$question)

class AnswerBusinessLogic private \$answerDao = null; private \$businessLogic = null; private \$maxTextLength = 60; public function __construct() public function getAllAnswers() public function getAnswerById(\$answerId) public function getAnswerByName(\$answerText) public function getAssocTags(\$answerld) public function addAnswer(Answer \$answer, \$tags) public function updateAnswer(Answer \$answer, \$tags) public function updatePosition(\$a1, \$a2) public function deleteAnswer (Answer \$answer) public function cleanAnswer(Answer \$answer) public function validateAnswer(Answer \$answer)

> class ProgramBusinessLogic private \$programDao = null; private \$businessLogic = null; private \$maxProgCodeLength = 10; private \$maxProgNameLength = 40; private \$maxProgLinkLength = 200; oublic function __construct() public function getAllPrograms() public function getAllProgramsAlphabetical() public function getProgramById(\$programId) public function getProgramByName(\$programName) public function addProgram(Program \$program, \$tags) public function updateProgram(Program \$program, \$tags public function deleteProgram(Program \$program) public function cleanProgram(Program \$program) public function validateProgram(Program \$program)

class TagBusinessLogic private \$tagDao = null; private \$businessLogic = null; private \$maxTagNameLength = 100;

public function __construct() public function getAllTags() public function getAllTagsAlphabetical() public function getTagById(\$tagId) public function getTagsByProgramId(\$programId) public function getAssocPrograms(\$tagId) public function addTag(Tag \$tag) public function updateTag(Tag \$tag) public function deleteTag(Tag \$tag) public function cleanTag(Tag \$tag)

public function validateTag(Tag \$tag)

TransferObject

class Admin

private \$adminId; private \$firstName; private \$lastName; private \$username; private \$password;

public function setAdminId(\$adm public function setFirstName(\$fi public function setLastName(\$I public function setUsername(\$ public function setPassword(\$) public function getAdminId() public function getFirstName() public function getLastName() public function getUsername() public function getPassword()

class Answer

private \$answerld: private \$answerText; private \$questionId;

public function setAnswerld(\$an public function setAnswerText(\$ public function setQuestionId(\$q public function getAnswerld() public function getAnswerText() public function getQuestionId()

class Department private \$departmentId;

private \$departmentCode; private \$departmentName;

public function setDepartmentId(\$departmentId)

public function setDepartmentCode(\$departmentCode) public function setDepartmentName(\$departmentName)

public function getDepartmentId() public function getDepartmentCode()

public function getDepartmentName()

class Program

private \$programId; private \$departmentId; private \$programCode; private \$programName; private \$programLink;

private \$counter; public function setProgramId(\$programId) public function setDepartmentId(\$departmentId) public function setProgramCode(\$programCode) public function setProgramName(\$programNam public function setProgramLink(\$programLink) public function modifyCounter(\$weight) public function getProgramId() public function getDepartmentId() public function getProgramCode()

public function getProgramName()

public function getProgramLink()

public function getCounter()

class Question private \$questionId: private \$questionText; public function setQuestionId(\$q public function setQuestionText(public function getQuestionId()

public function getQuestionText(

class Tag private \$tagld;

private \$tagText; public function setTagld(\$tagld) public function setTagText(\$tag public function getTagld() public function getTagText()

DataAccessLayer

AnswerDao

public function getResult(\$answerlds, \$weights, \$departmentld)

class CalculateResult

class DataSource private \$host private \$username

private \$password

admin

addadmin

addanswer

adddepartment

addprogram

addquestion

addtag

_ _ _ _ _ _ _ _ _ _ _ _

public function createConnection(\$dbName)

«Interface»

public function getAllAnswers();

public function getAnswerById(\$AnswerId); public function getAssocTags(\$Answerld);

public function addAnswer(Answer \$Answer, \$tags); public function updateAnswer(Answer \$Answer, \$tags); public function updatePosition(\$a1, \$a2); public function deleteAnswer(Answer \$Answer);

class AnswerDaoImpl implements AnswerDao

«Interface»

AdminDao

public function getAllAdmins();

public function getAdminByUser

public function addAdmin(Admin

public function updateAdmin(Ad

public function deleteAdmin(Adn

class AdminDaoImpl implements AdminDao

private \$dbName = "PSS";

public function getAllAnswers()

public function getAnswerById(\$answerId) public function getAnswerByText(\$answerText)

public function getAssocTags(\$answerld) public function addAnswer(Answer \$answer, \$tags) public function updateAnswer(Answer \$answer, \$tags)

public function updatePosition(\$a1, \$a2)

public function deleteAnswer(Answer \$answer)

private \$dbName = "PSS";

public function getAllAdmins()

public function getAdminById(\$adminId)

public function getAdminByUsername(\$username)

public function addAdmin(Admin \$admin, \$departments)

public function updateAdmin(Admin \$admin, \$departments)

viewdept

viewprogram

viewquestion

viewtag

questions

viewadmin

Viewanswer

answers

depts

login

public function getAssocDepartments(\$adminId)

public function deleteAdmin(Admin \$admin)

public function deleteDepartment(Department \$department)

«Interface»

ProgramDao

«Interface»

DepartmentDao

public function getDepartmentByCode(\$departmentCode);

public function addDepartment(Department \$department);

class DepartmentDaoImpl implements DepartmentDao

public function getAllDepartmentsAlphabetical();

public function getAssocAdmins(\$departmentId);

public function getDepartmentById(\$departmentId);

public function getAssocQuestions(\$departmentId);

public function getAssocPrograms(\$departmentId);

public function getAllDepartments();

public function getAllPrograms(); public function getAllProgramsAlphabetical();

public function getAdminById(\$a public function getProgramById(\$programId); public function getAssocDepartn

private \$dbName = "PSS";

public function getAllDepartments()

public function getAllDepartmentsAlphabetical()

public function getAssocAdmins(\$departmentId)

public function getAssocQuestions(\$departmentId)

public function getAssocPrograms(\$departmentId)

public function getDepartmentById(\$departmentId)

public function getDepartmentByCode(\$departmentCode)

public function addDepartment(Department \$department)

public function updateDepartment(Department \$department)

public function getProgramByName(\$programName); public function getAssocTags(\$programId);

public function addProgram(Program \$program, \$tags); public function updateProgram(Program \$program, \$tags);

public function deleteProgram(Program \$program);

private \$dbName = "PSS";

public function getAllPrograms() public function getAllProgramsAlphabetical() public function getProgramById(\$programId)

public function getProgramByName(\$programName) public function getAssocTags(\$programId)

public function addProgram(Program \$program, \$tags) public function updateProgram(Program \$program, \$tags) public function deleteProgram(Program \$program)

class TagDaoImpl implements TagDao class ProgramDaoImpl implements ProgramDao private \$dbName = "PSS";

public function getAllTags() public function getAllTagsAlphabetical()

public function getTagById(\$tagId) public function getAssocPrograms(\$tagld)

public function addTag(Tag \$tag)

«Interface» interface QuestionDao

public function addQuestion(Question \$question, \$departme

public function updateQuestion(Question \$question, \$depar

class QuestionDaoImpl implements QuestionDao

public function addQuestion(Question \$question, \$depart

public function updateQuestion(Question \$question, \$dep

public function updatePosition(\$departmentId, \$q1, \$q2)

«Interface»

interface TagDao

public function updatePosition(\$departmentId, \$q1, \$q2);

public function getQuestionById(\$questionId)

public function getAssocAnswers(\$questionId)

public function getQuestionByText(\$questionText)

public function getAssocDepartments(\$questionId)

public function deleteQuestion(Question \$question)

public function getTagsByProgramId(\$programId);

public function getAllQuestions();

private \$dbName = "PSS";

public function getAllTags();

public function getAllTagsAlphabetical();

public function getAssocPrograms(\$tagld);

public function getTagById(\$tagId);

public function addTag(Tag \$tag);

public function updateTag(Tag \$tag);

public function deleteTag(Tag \$tag);

public function getAllQuestions()

public function getQuestionById(\$questionId);

public function getQuestionByText(\$questionText);

public function getAssocDepartments(\$questionId);

public function deleteQuestion(Question \$question);

public function getAssocAnswers(\$questionId);

public function updateTag(Tag \$tag) public function deleteTag(Tag \$tag)

Student

Calvin Li Keith Sandy Gurbachan

TeeFour

