StudyUP
Software Architectural Design

Submitted to:

Asst. Prof. Ma. Rowena C. Solamo
Faculty Member
Department of Computer Science
College of Engineering
University of the Philippines, Diliman

Submitted by:
Ang, Karina Kylle L.
Kopio, Katrina Mae D.
Principio, Roberto Jr. D.

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2019-2020

Unique Reference:

The documents are stored in the project repository referenced with "StudyUP - Architectural Design" in GitHub with filename 'StudyUP -Architectural Design.pdf'.

Purpose:

In this document, all the classes in the workshops on User Interface Design, Data Design and Control classes are consolidated into a single software architecture for the Use Case Specifications of the group.

Audience:

The target audience are: Prof. Solamo, University of the Philippines Diliman Computer Science students, and anyone else interested in the development process of the StudyUP application.

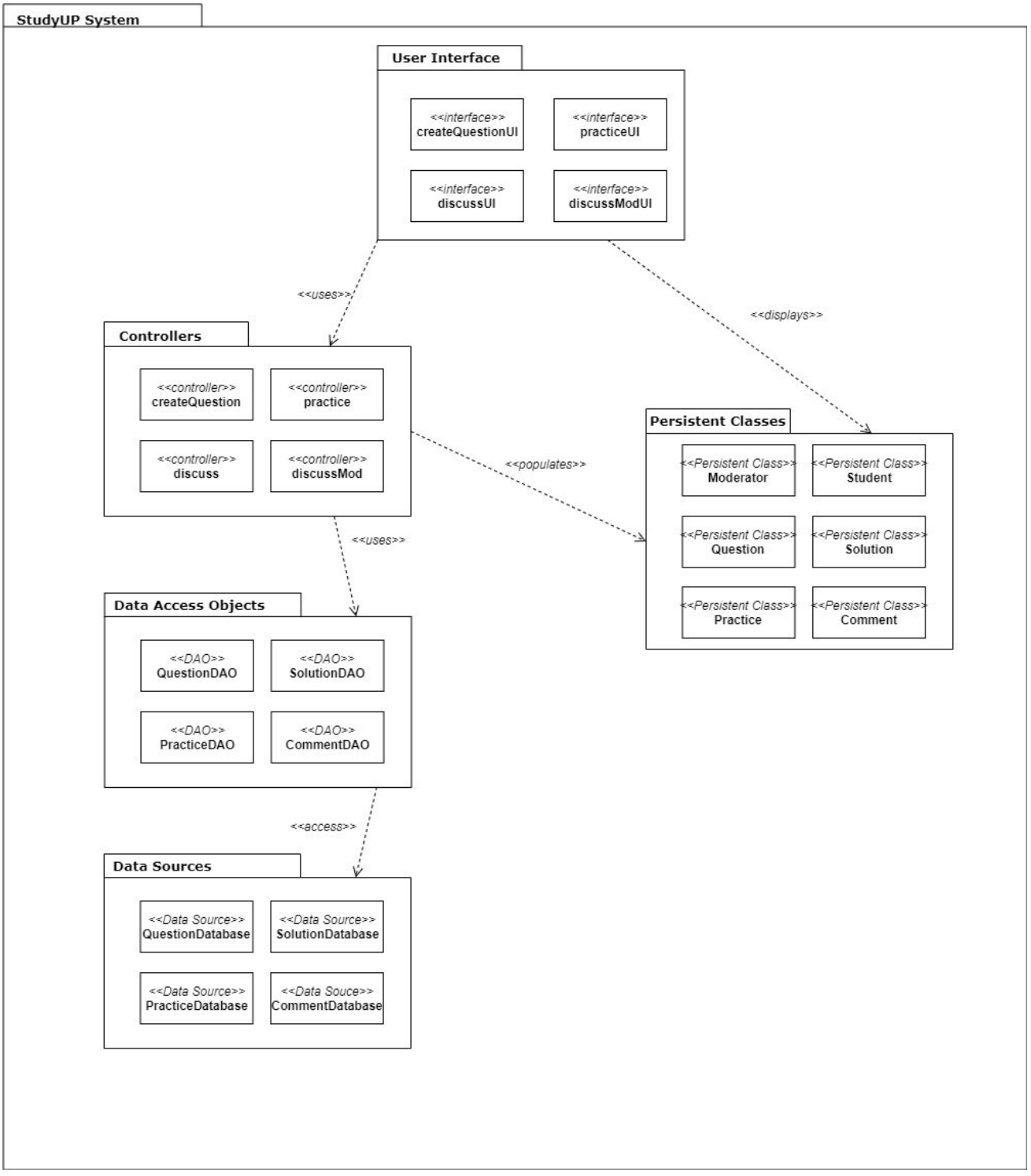
Revision Control:

Revision Date	*Person Responsible*	*Version Number*	*Contribution/Modification*
10/25/19	Roberto D. Principio Jr.	1.0	Initial Document
10/29/19	Roberto D. Principio Jr.	1.1	System Description, Data Sources, DAO attributes, Transfer Objects attributes
10/30/19	Karina Kylle L. Ang	1.2	Made Revised Software Architecture Model
10/30/19	Karina Kylle L. Ang	1.3	Added UI descriptions
10/30/19	Katrina Mae D. Kopio	1.4	Added UI Attributes and Responsibilities
10/30/19	Katrina Mae D. Kopio	1.5	Added Controller Description, Attributes and Responsibilities
10/30/19	Katrina Mae D. Kopio	1.6	Added DAO Attributes and Responsibilities
10/31/19	Karina Kylle L. Ang	1.7	Added Transfer Objects Methods
11/01/19	Karina Kylle L. Ang	1.8	Finalized document and tables

***System Name*:** StudyUP System

Description: StudyUP is a web application made by UP students for UP students. Its key feature is the testing module, where students can customize their own exercises and answer them. StudyUP fosters a collaborative learning environment as students can discuss and share tips, hints, and clarifications with other students about each and every problem.

***Revised Software Architecture Model*:**



User Interface Package:

Screen Name	**Description**
createQuestionUI	This is the screen of the moderator to the system whenever he or she needs to create a question. *Attributes*: String topic Question question Answer answer *Responsibilities*: makeQuestion (Question inputQuestion): void addQuestion (Question inputQuestion): void exitQuestion ( ): void addInformation (Date creationDate): void
practiceUI	This is the screen of the student to the system whenever he or she needs to practice. *Attributes*: Question question Answer answer int result *Responsibilities*: practice (Question inputQuestion): void getQuestion (String topic): int studentAnswer (Answer answer): void redirectToDiscuss ( ): void addInformation (int trialNo, Date examDate): void
discussUI	This is the screen of the student to the system whenever he or she needs to discuss with others. *Attributes*: Question question String studentComment *Responsibilities*: checkAnswer (Question input, Answer answer): void getStuAnswer ( ): int returnScore ( ): int addComment (String comment): void saveComment (String comment): void
discussModUI	This is the screen of the moderator to the system whenever he or she needs to discuss with others and do moderator duties. *Attributes*: Question question String studentComment String modComment *Responsibilities*: addModComment (String comment): void saveComment (String comment): void

	saveState (Forum forum): void removeStuComment (String comment): void addInformation (Date postDate): void

Controllers Package:

Controller Name	**Description**
createQuestion	This is the control that adds a created question into the system. *Attributes:* int QuestionID Date creationDate int timeLimit String topic String type Question question Answer answer *Responsibilities:* makeQuestion (Question inputQuestion): void addQuestion (Question inputQuestion): void exitQuestion ( ): void addInformation (Date creationDate): void
practice	This is the control that processes the student's answer to a question. *Attributes:* int StudentID Date dateTaken Question question Answer answer String topic String result int timeTaken int trialNo Date examDate *Responsibilities*: practice (Question inputQuestion): void getQuestion (String topic): int studentAnswer (Answer answer): void redirectToDiscuss ( ): void addInformation (int trialNo, Date examDate): void
discuss	This is the control that checks the answer given by the student, evaluates their score, and lets students participate in discussions by creating and commenting on posts. *Attributes:* Question question Answer studentAnswer String studentComment *Responsibilities:*

	checkAnswer (Question input, Answer answer): void getStuAnswer ( ): int returnScore ( ): int addComment (String comment): void saveComment (String comment): void
discussMod	This is the control that allows moderators to comment on posts in the discussion forum, or remove student comments. *Attributes:* Question question Date postDate Forum forum String studentComment String modComment *Responsibilities:* addModComment (String comment): void saveComment (String comment): void saveState (Forum forum): void removeStuComment (String comment): void addInformation (Date postDate): void

Data Access Objects Packages:

DAO Name	**Description**
QuestionDAO	This is the data access object *Question*, which contains the data about the question. *Attributes*: private Question theQuestion; private QuestionList theQuestionList; private Statement sqlStmnt private Connection con private ResultSet rs *Methods:* public void addQuestion (Question theQuestion)
SolutionDAO	This is the data access object *Solution*, which contains the data about the solution. *Attributes*: private Solution theSolution; private SolutionList theSolutionList; private Statement sqlStmnt private Connection con private ResultSet rs

	Methods: public void addSolution (Solution theSolution)
PracticeDAO	This is the data access object *Practice*, which contains the data about the practice data of a student.. *Attributes*: private Practice thePractice; private PracticeList thePracticeList; private Statement sqlStmnt private Connection con private ResultSet rs *Methods*: public void takePractice (Practice p)
CommentDAO	This is the data access object *Comment*, which contains the data about the comment. *Attributes*: private Comment theComment; private CommentList theCommentList; private Statement sqlStmnt private Connection con private ResultSet rs *Methods*: public void giveComment (Comment theComment)

Persistent Classes or Transfer Objects Package:

Class Name	**Description**
Moderator	This is the entity class *Moderator*, which contains the data about the moderator. *Attributes*: private int modID; private String name; *Methods*: public void addModerator (Moderator mod);
Student	This is the entity class *Student*, which contains the data about the student. *Attributes*: private int stuID; private String name;

	Methods: public void addStudent (Student stu);
Question	This is the entity class *Question*, which contains the data about the question. *Attributes*: private int questionID; private Varchar courseID; private int unitNo; private int chapNo; private Varchar images; private Varchar question; private choiceA; private choiceB; private choiceC; private choiceD; private int solutionID; private int threadID; *Methods*: public void addQuestion (Question q);
Solution	This is the entity class *Solution*, which contains the data about the solution. *Attributes*: private int solutionID; private int questionID; private Varchar answer; private Varchar solution; private Varchar images; *Methods*: public void addSolution (Solution s);
Practice	This is the entity class *Practice*, which contains the data about the practice data of a student.. *Attributes*: private int examSessionID; private int userID; private int questionID; private Varchar userAnswer;

	Methods: public void takePractice (Practice p);
Comment	This is the entity class *Comment*, which contains the data about the comment. *Attributes*: private int commentID; private int threadID; private Datetime date; private int userID; private Varchar comment; private int replyID; *Methods*: public void giveComment (Comment c);

Data Sources Package:

File Name or Database Name	**Description**
QuestionDatabase	This is the data source of questions found in a relational database system.
SolutionDatabase	This is the data source of solutions found in a relational database system.
PracticeDatabase	This is the data source of practice data found in a relational database system.
CommentDatabase	This is the data source of comments found in a relational database system.