RUPE
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:
Dylan Bayona
Annysia Dupaya
Makki Villaluz

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2019-202

Unique Reference:

The documents are stored in the following link: https://github.com/chinadupaya/RUPE/tree/master/03-Design%20Engineering referenced with the following filename: 'RUPE - Architectural Design.pdf'.

Purpose:

The purpose of this document is to consolidate all of the classes that were elaborated in our previous workshops involving User Interface Design, Data Design, and Control classes into a single software architecture for the Use Case Specification. Related classes will be grouped together using package diagrams. The objective of this document is to help the developers produce a portion of the Revised Software Architecture.

Audience:

The audience of this document includes the developers of the RUPE application for reference and anyone who wishes to review their work for analysis and review. It also includes anyone who wishes to track the developers' progress for the course CS 191 - Software Engineering I.

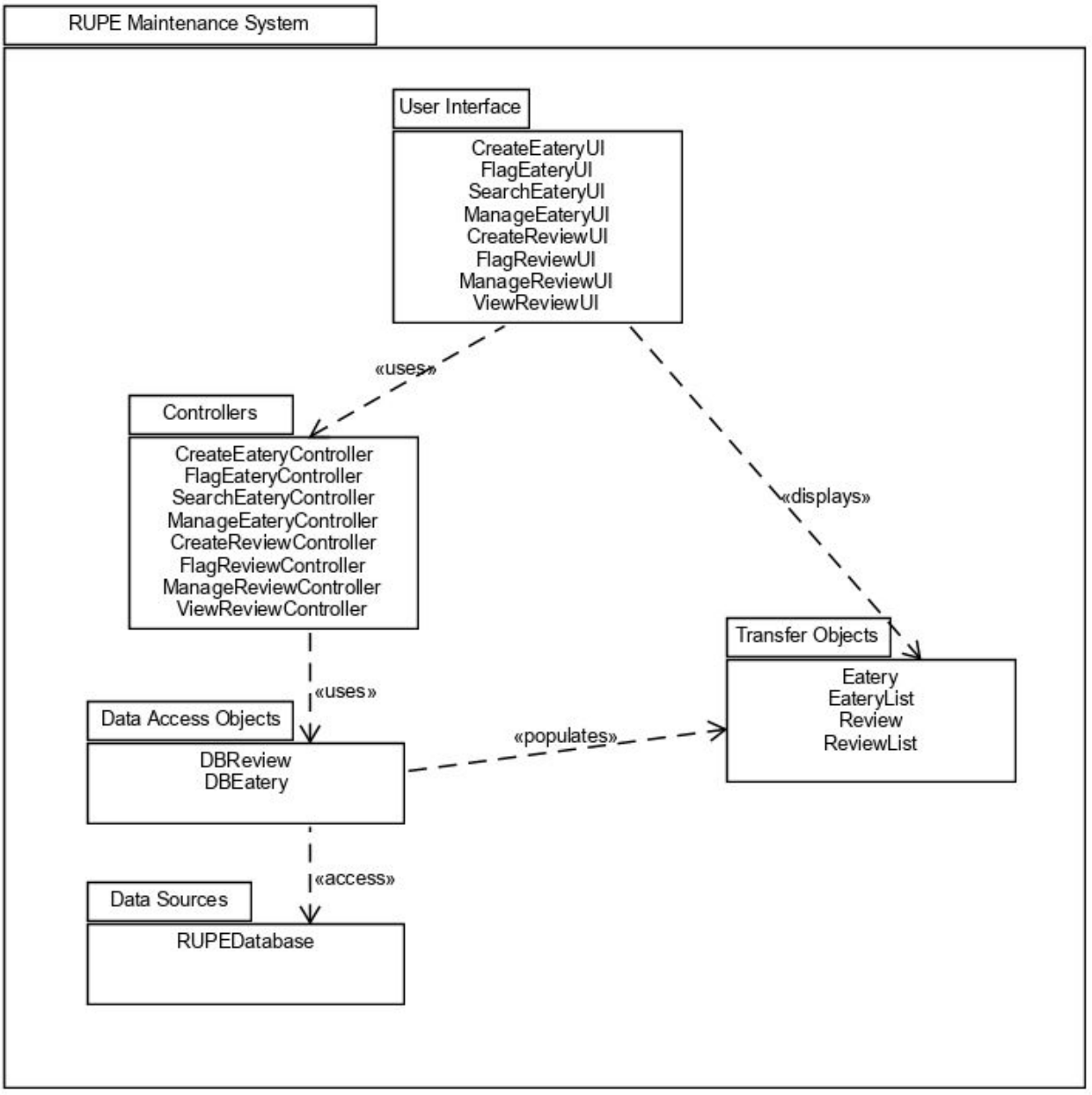
Revision Control:

Revision Date	*Person Responsible*	*Version Number*	*Contribution/Modification*
10/25/2019	Annysia Dupaya	1.0	Initial Document; Version number should match the one found in the footer.
10/29/2019	Dylan Bayona	1.1	Added document unique reference, document purpose, and document audience; also added Transfer Object Packages
10/29/2019	Annysia Dupaya	1.2	Made the Architecture Model
10/29/2019	Dylan Bayona	1.3	Added UpdateEateryController under Controller Packages
10/30/2019	Dylan Bayona	1.4	Added system description and Data Sources Packages
10/30/2019	Annysia Dupaya	1.5	Added User Interface and Controller Packages
11/1/2019	Makki Villaluz	1.6	Added Data Access Objects and User Interface Packages

***System Name*:** RUPE Maintenance System

Description: This system is centered on providing a review system for the various eateries found in UP Diliman. Reviewers can rate and review the food of these eateries. They can also search for eateries, view eateries, view food reviews made by other reviewers, flag inappropriate reviews, and flag eateries which are closed. Administrators can view food reviews and eateries, and they can also manage food reviews and eatery records. In the case of an inappropriate food review or a flagged eatery, they can modify or delete these records accordingly.

***Revised Software Architecture Model*:**



User Interface Package:

Screen Name	**Description**
CreateEateryUI	This is the interface of the reviewers to the system when they add a new eatery that does not currently exist in the application. *Responsibilities*: *[EateryDoesNotExist]] createNewEatery(String newName, String newAddress, String newContact, String newStatus) : void
FlagEateryUI	This is the interface of the reviewers to the system when they find an eatery that does not exist in real life. *Responsibilities*: *[hasBeenFlagged] flagEateryIO(int newIsFlagged) : void
SearchEateryUI	This is the interface of the reviewers to the system when they want to look for an eatery in the application. *Responsibilities*: viewEateries(String s, Eatery[] e) : void
ManageEateryUI	This is the interface of the administrators to the system when an eatery has been flagged and needs to be deleted or updated. *Responsibilities*: *[ConfirmedDeletion] deleteEateryIO() : void updateEateryIO(Eatery e) : void
CreateReviewUI	This is the interface of the reviewers to the system when they add a new review to an eatery. *Responsibilities*: createNewReview(String newText, float newRating) : void
FlagReviewUI	This is the interface of the reviewers to the system when they find a review that is malicious or unhelpful. *Responsibilities*: flagReviewIO(int newIsFlagged) : void
ManageReviewUI	This is the interface of the administrators to the system when a review has been flagged and needs to be deleted. *Responsibilities*: *[ConfirmedDeletion] deleteReviewIO() : void updateReviewIO(Eatery e, Review r) : void

ViewReviewUI	This is the interface of the reviewers to the system when a review has been flagged and needs to be deleted. *Responsibilities*: *[ReviewExists, notFlagged] viewReviewIO(Review r) : void

Controllers Package:

Controller Name	**Description**
ManageEateryController	This is the control that lets the administrator update an eatery's details. *Attributes*: private Eatery e; *Responsibilities*: *[doesEateryExist = no] doNothing(Eatery e) : void *[didEateryClose = yes] deleteEatery(Eatery e) deleteEatery(Eatery e) *[didEateryMove = yes OR didContactChange = yes]updateEateryDB(Eatery e) : void
CreateEateryController	This is the control that handles the creation of a new eatery. *Responsibilities*: *[sufficientInfoInBoundary] addNewEatery(Eatery e) : void
FlagEateryController	This is the control that flags an eatery. *Responsibilities*: FlagEatery(Eatery e) : int
SearchEateryController	This is the control that searches for an eatery in the application. *Responsibilities*: searchEatery(String s) : void
CreateReviewController	This is the control that handles the creation of a food review. *Responsibilities*: *[reviewerCreatesReview] createReview(Review r) : void
FlagReviewController	This is the control that flags a malicious food review. *Responsibilities*: *[reviewMayContainMaliciousContent] FlagReview(Review r) : void
ManageReviewController	This is the control that deletes a food review. *Responsibilities*: deleteReview(Review r) : void *[changeToEatery] updateReviewList(Eatery e, Review r) : void
ViewReviewController	This is the control that shows a food review.

	Responsibilities: *[ReviewExists, notFlagged] viewReview(Review r)

Data Access Objects Packages:

DAO Name	**Description**

DBReview	This data access object is responsible for getting the review data from the database
	Attributes:
	private Review r;
	private Connection con;
	private Statement sqlStmt;
	private ResultSet rs;
	Methods:
	private String prepareInsertStmt(Review r);
	private String prepareDeleteStmt(Review r);
	private String prepareUpdateStmt(Review r);
	public int addReview(Review r);
	public int deletereview(Review r);

DBEatery	This data access object is responsible for getting the eatery data from the database
	Attributes:
	private Eatery e;
	private Connection con;
	private Statement sqlStmt;
	private ResultSet rs;
	Methods:
	private String prepareSelectStmt(String searchCriteria);
	private String prepareInsertStmt(Eatery e);
	private String prepareDeleteStmt(Eatery e);
	private String prepareUpdateStmt(Eatery e);
	public int findEatery(String searchCriteria);
	public int addEatery(Eatery e);
	public int editEatery(Eatery e);
	public int deleteEatery(Eatery e);

Transfer Objects Package:

Class Name	Description

Review	This represents the entity class FoodReview, which contains information about a review for a certain eatery.
	Attributes:
	private int ReviewID;
	private String Text;
	private bool isFlagged;
	private float rating;
	private int EateryID;
	Methods:
	private void setReviewID(); // no parameters passed since ReviewID is auto-incremented here
	private void setText(String Text);
	private void setReviewFlag(bool isFlagged);
	private void setReviewRating(float rating);
	private void setEateryID(int EateryID);
	private void getReviewID();
	private void getText();
	private void getReviewFlag();
	private void getReviewRating();
	private void getEateryID();

ReviewList	This represents a list of Review entities.
	Attributes:
	private Review[] ReviewList;
	Methods:
	private void add(Review r);
	private void set(int index, Review r);
	private void get(int index);

Eatery	This represents the entity class Eatery, which contains information about a particular eatery. *Attributes:* private int EateryID; private String Name; private String Address; private String Contact; private bool isFlagged; private float rating;

	private Review[] ReviewList;
	Methods:
	private void setEateryID(); // no parameters passed since EateryID is auto-incremented here
	private void setName(String Name);
	private void setAddress(String Address);
	private void setContact(String Contact);
	private void setEateryFlag(bool isFlagged);
	private void setEateryRating(float rating);
	private void getReviewID();
	private void getName();
	private void getAddress();
	private void getContact();
	private void getEateryFlag();
	private void getEateryRating();

EateryList	This represents a list of Eatery entities.
	Attributes:
	private Eatery[] EateryList;
	Methods:
	private void add(Eatery e);
	private void set(int index, Eatery e);
	private void get(int index);

Data Sources Package:

File Name or Database Name	Description
RUPEDatabase	This is the data source of eatery and review records found in a relational database system. It contains all of the eateries as well as their corresponding reviews. The database contains two (2) tables: the Eatery and Review tables.