SARAPP
(Search and Rate Application
of Food Places in UP Diliman)
Analysis Model

Submitted to:

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

Submitted by:

Angelika Galang
Julian Troy Valdez
Richelle Yap

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2017-2018

Revision Control

History Revision:

Revision Date	Person Responsible	Version Number	Modification
10/22/17	Richelle Yap	1.0	Initial Document
10/26/17	Julian Troy Valdez	2.0	Added Entity and Control class entries
10/27/17	Angelika Galang	3.0	Added Boundary classes and Class diagram; Edited Control classes; Edited Entity classes
11/27/17	Angelika Galang	3.1	Edited project name, analysis diagram, RateFoodStoreUI description, and entity classes' name and description-- all for consistency with other files

Purpose:

This Document illustrates the Analysis model of the projects' Requirements specification
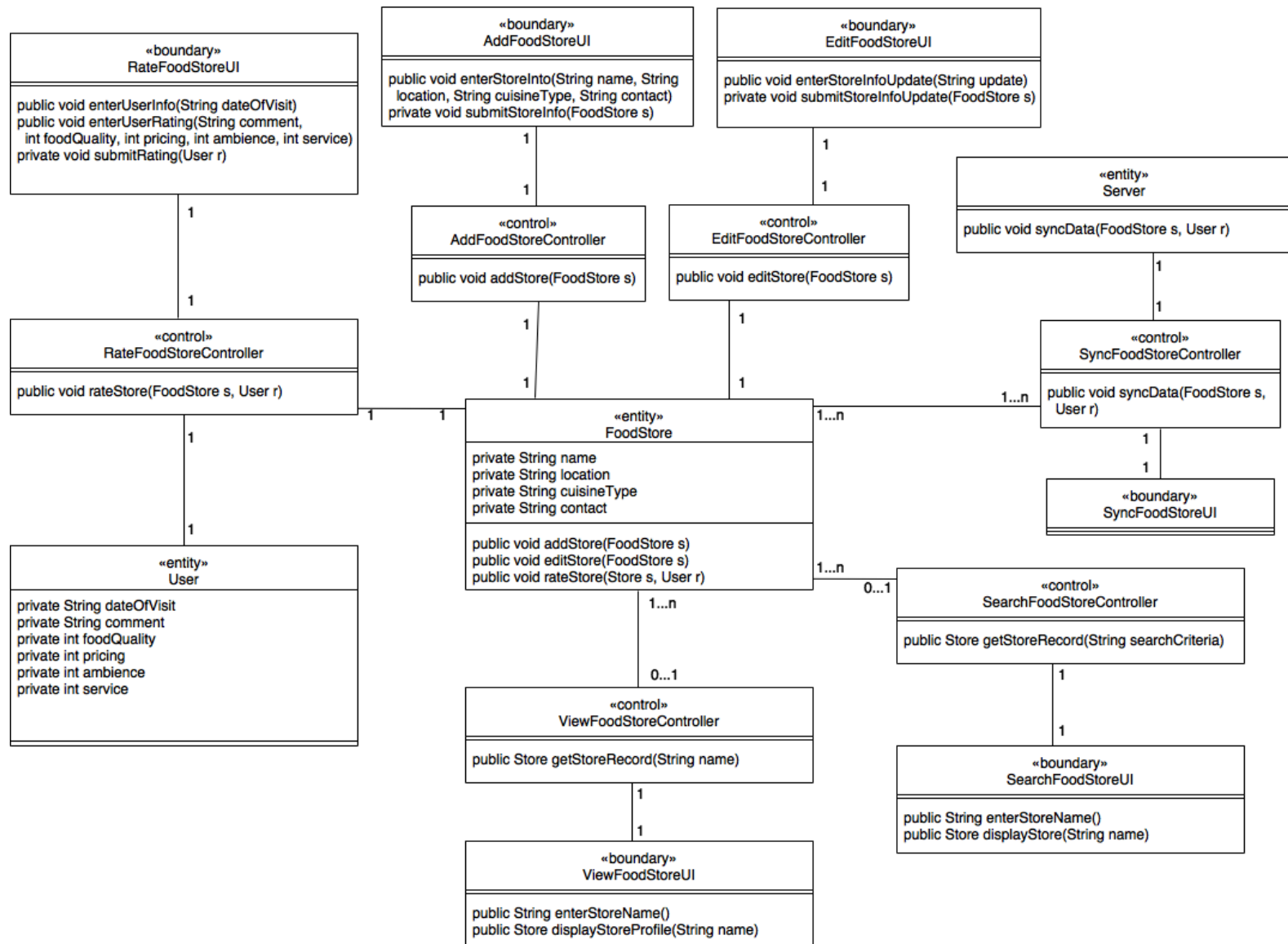
Audience:

This document serves as a partial fulfillment of academic requirements for the CS 191 Software Engineering course, handled by Ma'am Rowena Solamo, to whom this document is made for. Additionally, the users of the application will mainly consist of students, teachers, staff, and guests who would visit and look for some place to eat inside University of the Philippines Diliman.

System Name: SARAPP system

Description: SARAPP is a mobile application that allows students to rate and search for food places around University of the Philippines Diliman. This application helps students look and discover food places that match their needs according to budget, location, ambiance, and food quality. This also allows food place owners to see the ratings to deliver to the customers' needs.

Class Diagram:

«boundary» RateFoodStoreUI
- public void enterUserInfo(String dateOfVisit)
- public void enterUserRating(String comment, int foodQuality, int pricing, int ambience, int service)
- private void submitRating(User r)

«boundary» AddFoodStoreUI
- public void enterStoreInto(String name, String location, String cuisineType, String contact)
- private void submitStoreInfo(FoodStore s)

«boundary» EditFoodStoreUI
- public void enterStoreInfoUpdate(String update)
- private void submitStoreInfoUpdate(FoodStore s)

«entity» Server
- public void syncData(FoodStore s, User r)

«control» AddFoodStoreController
- public void addStore(FoodStore s)

«control» EditFoodStoreController
- public void editStore(FoodStore s)

«control» RateFoodStoreController
- public void rateStore(FoodStore s, User r)

«control» SyncFoodStoreController
- public void syncData(FoodStore s, User r)

«entity» FoodStore
- private String name
- private String location
- private String cuisineType
- private String contact

- public void addStore(FoodStore s)
- public void editStore(FoodStore s)
- public void rateStore(Store s, User r)

«boundary» SyncFoodStoreUI

«entity» User
- private String dateOfVisit
- private String comment
- private int foodQuality
- private int pricing
- private int ambience
- private int service

«control» SearchFoodStoreController
- public Store getStoreRecord(String searchCriteria)

«control» ViewFoodStoreController
- public Store getStoreRecord(String name)

«boundary» SearchFoodStoreUI
- public String enterStoreName()
- public Store displayStore(String name)

«boundary» ViewFoodStoreUI
- public String enterStoreName()
- public Store displayStoreProfile(String name)

Boundary Classes:

Class Name	Description
SearchFoodStoreUI	This interface lets the User search for a specific food store by inputting its name.
RateFoodStoreUI	This interface allows the User to rate a food store. Users can enter feedback of the store's food quality, pricing, ambience, and service. Users can also opt to add pictures and comments. In this interface, users have the option to add their date of visit.
ViewStoreProfileUI	This interface contains the food store's information which includes its name, location, cuisine type, picture (if available), contact details, and ratings submitted by the users (if there are any).
AddFoodStoreUI	This interface permits the Admin to add another food store with its corresponding information to the current existing list of food stores.
EditFoodStoreUI	This interface permits the Admin to edit an existing food store's information.
SyncFoodStoreSI	This system interface syncs the data available to the User and Admin to ensure unambiguity of the latest updated information about the food stores.

Control Classes:

Class Name	Description
SearchFoodStoreController	This control class can be performed by the User to search a food store from the list of existing food stores.
RateFoodStoreController	This control class allows the User to add a rating to a food store.
ViewFoodStoreController	This control class allows the both the User and the Admin to view the food stores' available information.
AddFoodStoreController	This control class is responsible for letting the Admin add a new instance of a food store along with the necessary information.
EditFoodStoreController	This control class lets the Admin provide new information about the food store to replace previous existing food store information.
SyncFoodStoreController	This control class is performed by the Server to sync data to ensure that information found in the web application is the same as found in the mobile and vice versa.

Entity Classes:

Class Name	Description
User	This entity contains the user's date of visit together with his/her rating of the food store that includes comments and numerical rating of the store's food quality, pricing, ambience, and service.
FoodStore	This entity contains a store's information which includes its name, location, cuisine type, contact details, and ratings added by users. The Admin uses this entity to add and edit a store's data from the database.
Server	The entity class responsible for syncing the data from the web application to the mobile and vice versa for consistency.