Iskolivery: A Crowdsourced Courier Service
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:
Fernandez, Aleksei Dominic C.
Legaspi, Bridget Noelle C.
Teves, Marc

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering
of the
1st Semester, AY 2018-2019

Unique Reference:

The documents are stored in https://github.com/marcteves/cs191-project/wiki/Project-Deliverables referenced with Iskolivery - Architectural Design.pdf under the section of 03 - Design Engineering.

Purpose:

This document is provided to develop a portion of the Revised Software Architecture.

Audience:

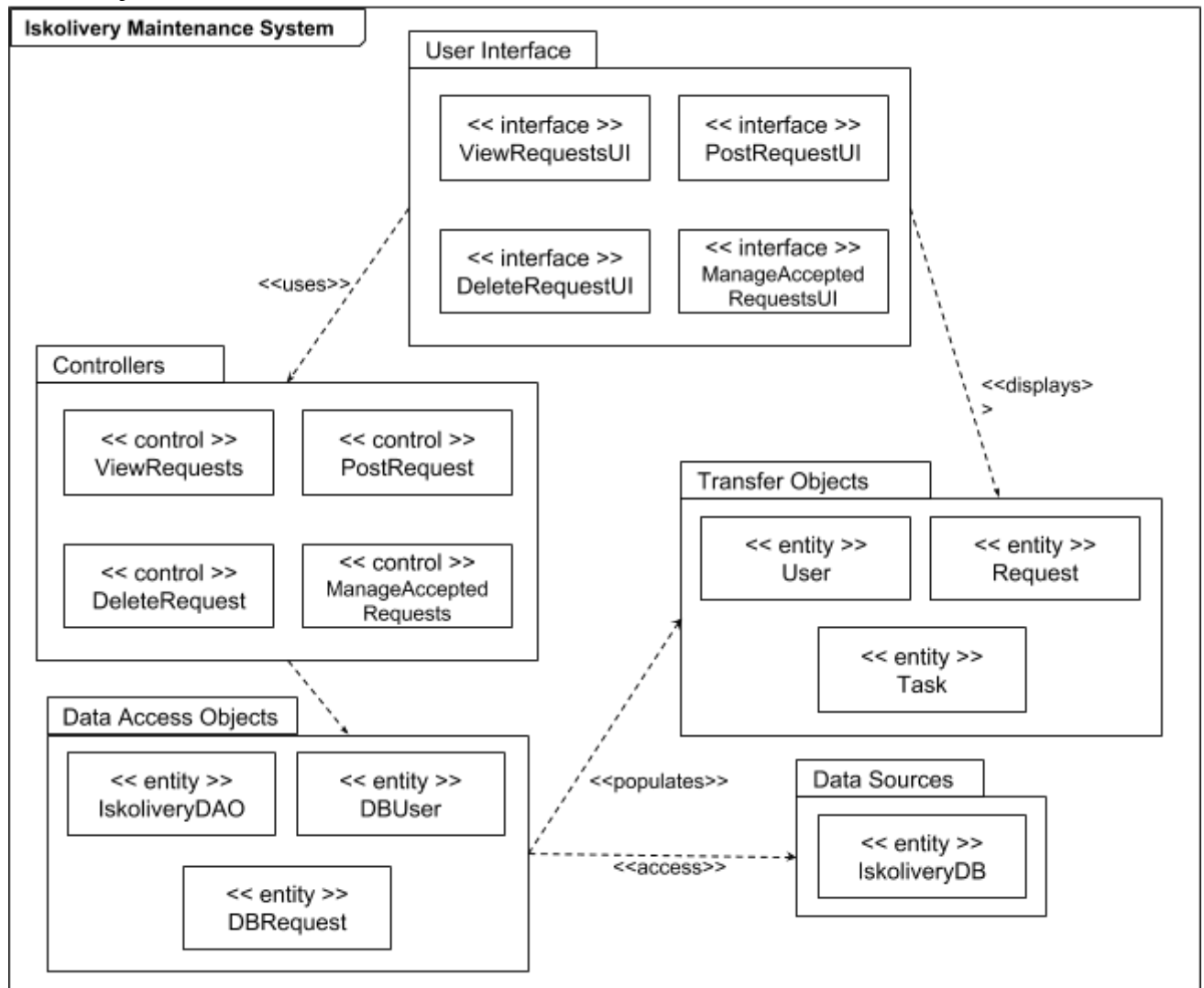Students and faculty of the University of the Philippines Diliman.

Revision Control:

Revision Date	Person Responsible	Version Number	Contribution/Modification
11/01/2018	Bridget Noelle Legaspi	1.0	Prepared the initial document, added UI package, controllers package, transfer objects package, and data source package.
11/06/2018	Bridget Noelle Legaspi	1.1	Added the software architecutre model
11/06/2018	Marc Teves	1.2	Added description of DAO and DB classes

System Name: Iskolivery Maintenance System

Description: Iskolivery connects people who need items or services (eg. photocopying, buying school supplies, car ride, tutorial, etc.) to the people who can deliver these items or services for them. In the application, requesters will be able to post a request and fulfillers may accept these requests. This maintenance system has four interfaces in UI package, four controls in Controllers package, three entities in Data Access Objects and Transfer Objects, and one entity for Data Sources package.

Revised Software Architecture Model:

User Interface Package:

Screen Name	Description
ViewRequestsUI	This is the interface for viewing and selecting requests. *Responsibilities*: public void ViewRequests(Integer usernameID) public void SelectRequest(Integer requestID)
PostRequestUI	This is the interface for posting new requests. *Responsibilities*: public void PostRequest(Integer usernameID, ArrayList<Task> requestInfo, Time deliveryTime, String deliveryLocation)
DeleteRequestUI	This is the interface for deleting a posted request. *Responsibilities*: public void DeleteRequest(Integer requestID)
ManageAcceptedRequestsUI	This is the interface for the fulfiller to manage their own accepted requests (ie. update, delete, add a request) *Responsibilities*: public void viewAcceptedRequests(username, usernameID)

Controllers Package:

Controller Name	Description
ViewRequests	This is the control that allows a user to view requests. *Responsibilities*: public void ViewUserRequests(Integer usernameID) public void SelectUserRequest(Integer requestID)
PostRequest	This is the control that allows a user to post requests. *Responsibilities*: public void PostRequest(Integer usernameID, ArrayList<Task> requestInfo, Time deliveryTime, String deliveryLocation) public void ValidateRequest(Integer usernameID, ArrayList<Task> requestInfo, Time deliveryTime, String deliveryLocation)
DeleteRequest	This is the control that allows a user to delete a request they posted. *Responsibilities*: public void DeleteUserRequest(Integer requestID)
ManageAcceptedRequests	This is the control that allows a user to manage their accepted requests (ie. update, delete, add) *Responsibilities*: public void viewAcceptedRequest(String username, Integer usernameID, Integer requestID)

Data Access Objects Packages:

DAO Name	Description

IskoliveryDAO	This data access object is responsible for getting user/request data from a file called IskoliveryDB.
	
	 Attributes:
	
	 Methods:
	
	 private void connectIskoliveryDatabase(String URL, String username, String password);
	
	 private DBUser createDBUser(String query);
	
	 private DBRequest createDBRequest(String query);

DBUser	This data access object is responsible for getting user data from a database.
	
	 Attributes:
	
	 private User user;
	
	 Methods:
	
	 private void insertUser(User user);
	
	 private void updateUser(User user);
	
	 private void deleteUser(User user);

DBRequest	This data access object is responsible for getting request data from a database.
	
	 Attributes:
	 private Request request;
	
	 Methods:
	
	 private void insertRequest(Request request);
	
	 private void updateRequest(Request request);
	
	 private void deleteRequest(Request request);

Transfer Objects Package:

Class Name	Description
User	This is the entity class User, which contains the data about the user. Admin is just a special instance of User. *Attributes*: private int usernameID public String username private String upmail private String college private String course private String location private float rating *Methods*: public void ViewUserAccountDetails(Integer usernameID, String username, String upmail, String college, String course, String location, Integer rating, Boolean disableToggle, Boolean verifyToggle)
Request	This is the entity class Request, which contains the information about the request of the user. *Attributes*: public User requester private User fulfiller private ArrayList<Task> requestInfo private Time deliveryTime private String deliveryLocation private int requestStatus
Task	This is the entity class Task, which contains the data about the tasks. This contains the actual information of the request of user. *Attributes*: private String requestInfo private String sourceLocation private int taskStatus

Data Sources Package:

File Name or Database Name	Description
IskoliveryDB	This database contains the data used by the Iskolivery system.