UPLog
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:
Ang, Aira Jazel
Bueno, Patricia Bianca
Valdez, Arian Allenson

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 https://github.com/Secretmapper/OrgUP referenced with Group 7-UPLog-Analysis Model.

Purpose:

To serve as a blueprint of the software the group needs to design and implement

Audience:
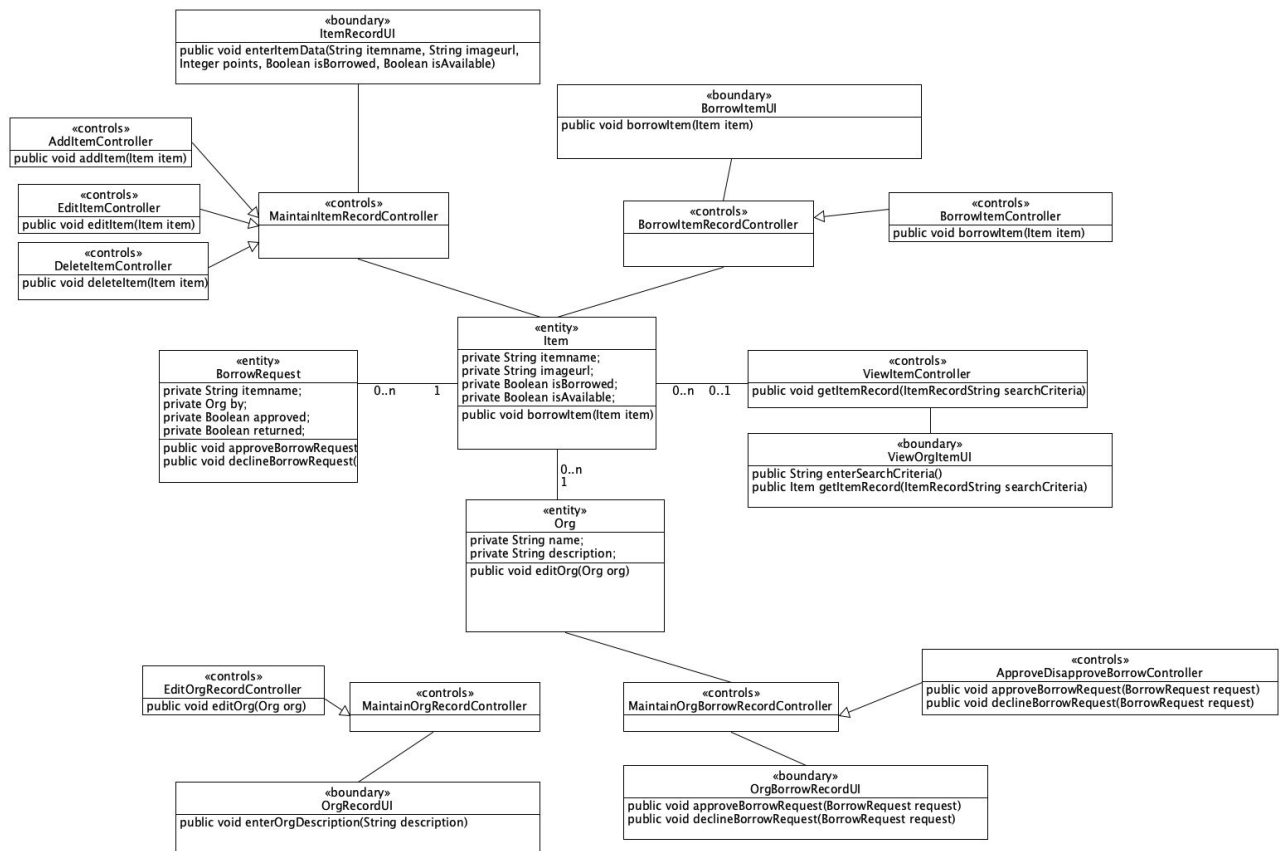
Any organization within UP Diliman

Revision Control:

Revision Date	Person Responsible	Version Number	Modification
10/01/19	Aira Jazel Ang	1.0	Initial Document
10/02/19	Arian Allenson Valdez	2.0	Add Analysis Model Diagram, Behaviour Model Diagrams for scenarios, add select Classes
10/02/19	Bianca Bueno	3.0	Added Purpose and Use Case name and description

***System Name*:**   UPLog - Manage Org Description/Items

***Description*:**   This use-case will allow org administration to manipulate their records such as their name, description, and the items they can lend to other organizations. This also allows org administration to add, edit, and delete items on their initial listed items. The borrowed/received items can also be viewed and accessed through this use case.

***Analysis Model*:**

Boundary Classes:

Class Name	Description
ItemRecordUI	This is the interface of the admin/user to the system whenever he or she needs to maintain item records. *Responsibilities*: public void enterItemData(String itemname, String imageurl, Integer points, Boolean isBorrowed, Boolean isAvailable) public void enterOrgDescription(String description)
OrgRecordUI	This is the interface of the admin/user to the system whenever he or she needs to maintain org record. *Responsibilities*: public void enterOrgDescription(String description)

Control Classes:

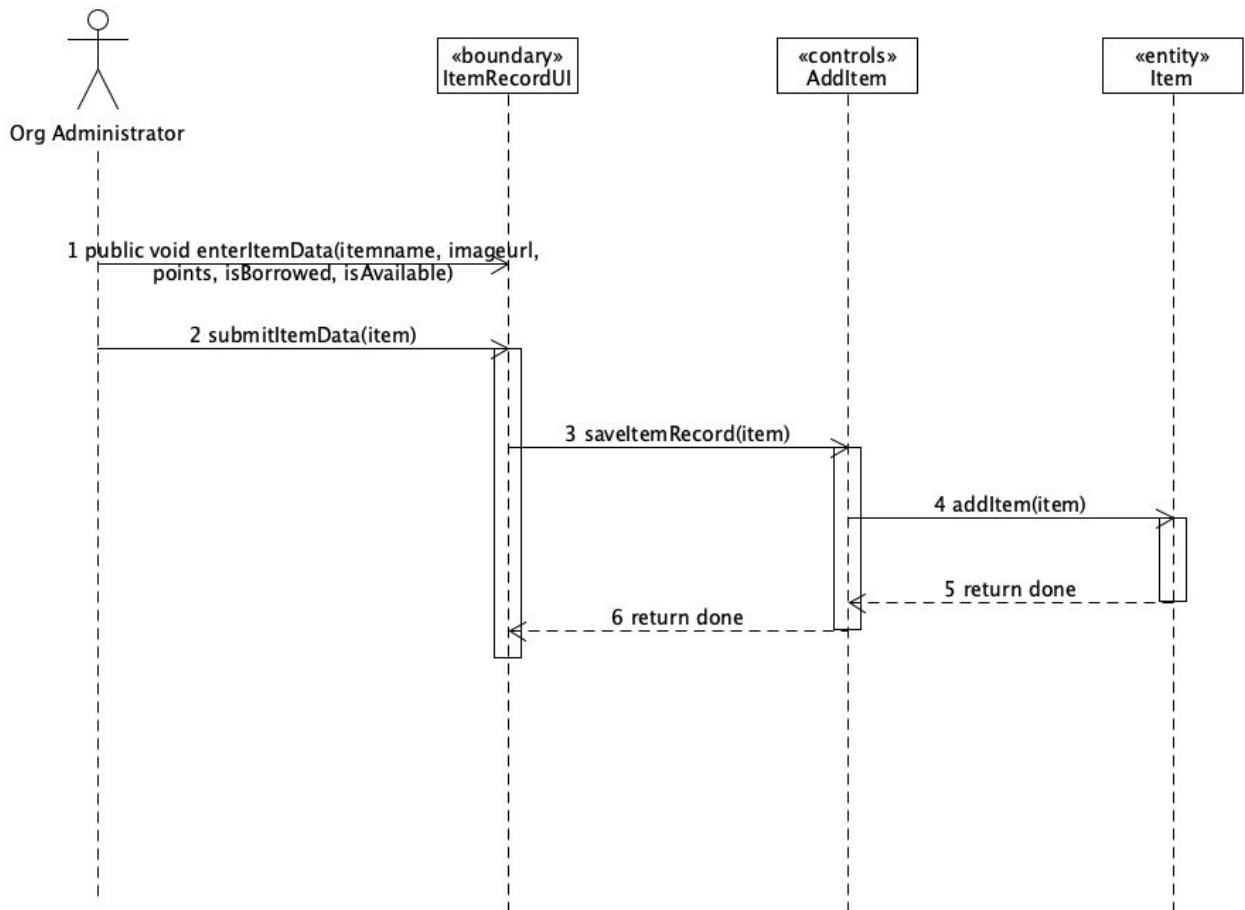| Class Name	Description
MaintainItemController	This is the control that maintain item records. It is considered an abstract class.
AddItemController	This is the control that adds an item to the system. It extends MaintainItemController. *Responsibilities*: public void AddItem(Item item)
EditItemController	This is the control that edits an item in the system. It extends MaintainItemController. *Responsibilities*: public void editItem(Item item)
DeleteItemController	This is the control that deletes an item in the system. It extends MaintainItemController. *Responsibilities*: public void deleteItem(Item item)
MaintainOrgRecordController	This is the control that maintain org records. It is considered an abstract class.
EditOrgRecordController	This is the control that adds info to orgs to the system. It extends MaintainOrgController. *Responsibilities*: public void editOrg(Org org)

Entity Classes:

Class Name	Description
Item	This is the entity class item, which contains the data about items borrowable. *Attributes*: private String itemname; private String imageurl; private Boolean isBorrowed; private Boolean isAvailable;
Org	This is the entity class org, which contains the data about the org *Attributes*: private String name; private String description;

Behavioral Model:

Use-Case Name: 1.0 Manage Org Record

Description: This use-case will allow org administration to manipulate their records such as their name, description, and the items they can lend to other organizations. This also allows org administration to add, edit, and delete items on their initial listed items. The borrowed/received items can also be viewed and accessed through this use case.
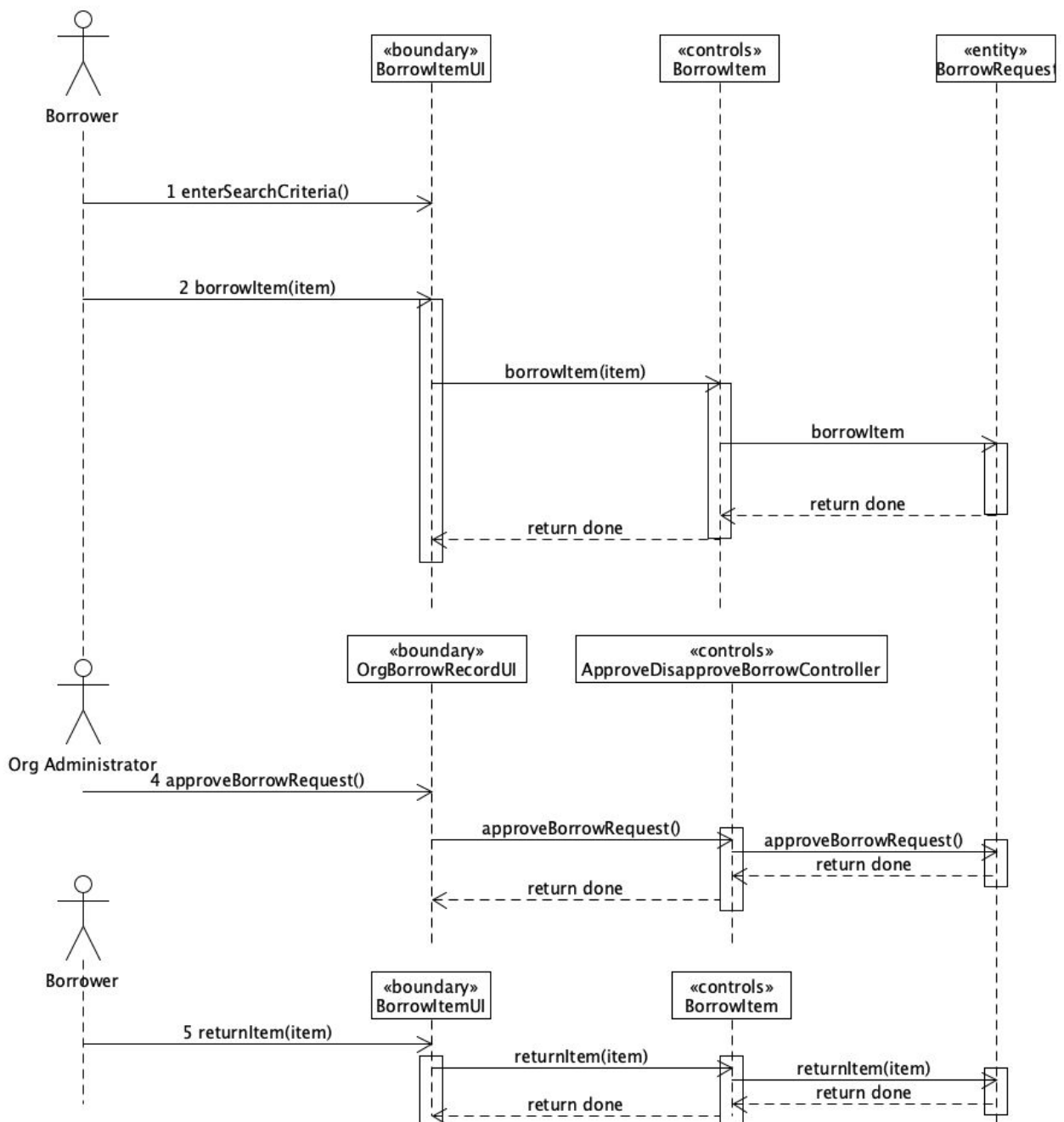
Scenario 1: Add item record successfully. (Basic Flow)

Use-Case Name: 1.0 Manage Org Record

Description: This use-case will allow org administration to manipulate their records such as their name, description, and the items they can lend to other organizations. This also allows org administration to add, edit, and delete items on their initial listed items. The borrowed/received items can also be viewed and accessed through this use case.
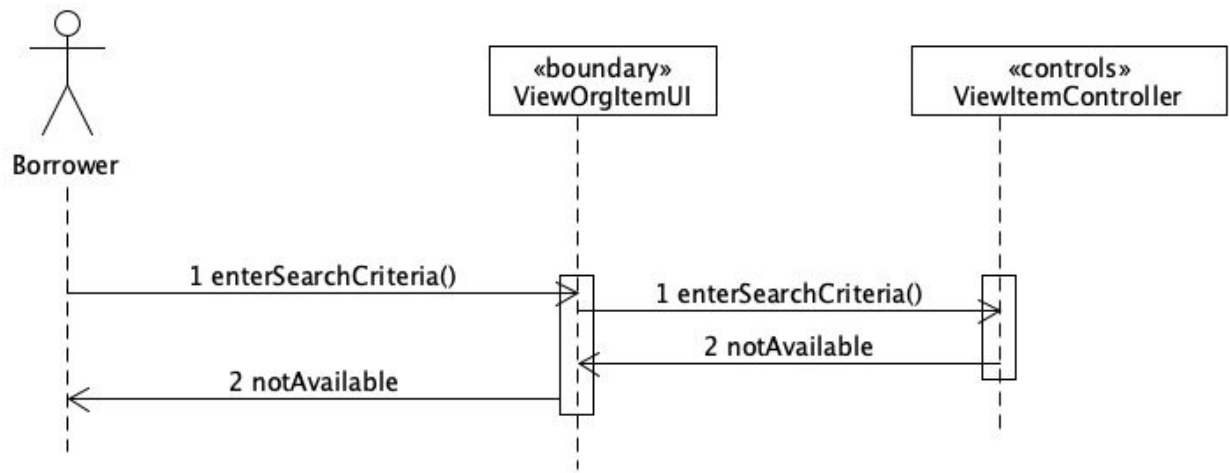
Scenario 1 (Basic Flow)

Use-Case Name: 2.0 Borrow

Description: This use-case will allow non-organization administrators and org administrators borrow certain items from organization administrators to be used in their event. Users can also view the borrowed list.

Scenario 2

Use-Case Name: 2.0 Borrow

Description: This use-case will allow non-org administrators and org administrators borrow certain items from organization administrators to be used in their event. Users can also view the borrowed list. Org Administrator can decide if he will allow or decline the borrowing request.

Scenario 3