

MaroonPrint

Software Architectural Design

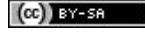
Submitted to:

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

Submitted by:

Lee, Kristine-Clair
Magno, Hannah Mae
Wu, Jeremy Jin Qian

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



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Unique Reference:

The documents are stored in the <https://maroonprint.tumblr.com/project-deliverables> referenced with MaroonPrint - Architectural Design.

Purpose:

The purpose of this document is to define the consolidated classes from the previous documents to help the developers finalize the software architecture of the system.

Audience:

University of the Philippines Diliman engineering students, faculty, and other personnel and also people who are assigned in maintaining the fire exits.

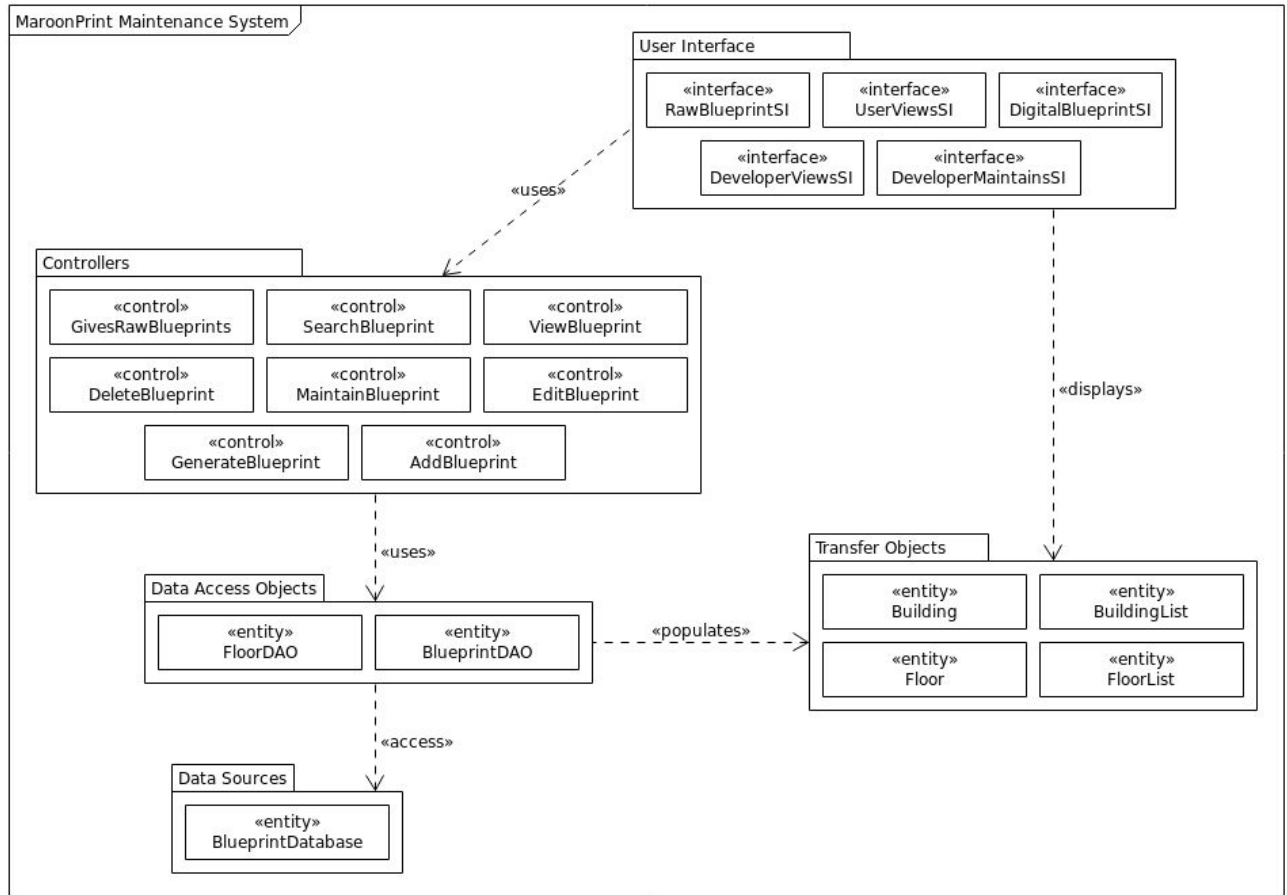
Revision Control:

<i>Revision Date</i>	<i>Person Responsible</i>	<i>Version Number</i>	<i>Contribution/Modification</i>
11/02/2018	Kristine-Clair Lee	1.0	Initial Document; Added diagram; Added contents for transfer object packages
11/02/2018	Hannah Mae Magno	1.1	Added the contents for controller and data sources package
11/02/2018	Wu, Jeremy Jin Qian	1.2	Added contents for User Interface package. Edited format.
02/06/2019	Hannah Mae Magno	1.3	Changed the contents for Data Access Objects Packages (DAO)
02/20/2019	Kristine-Clair Lee	1.4	Updated UML diagram and entities

System Name: MaroonPrint Maintenance System

Description: There are 5 boundary classes in the analysis model, namely, DeveloperMaintainSI, DeveloperViewUI, UserViewsUI, RawBlueprintUI, and DigitalBlueprintSI. In addition, there are 8 control classes which are GenerateBlueprint, SearchBlueprint, ViewBlueprint, GiveRawBlueprint, MaintainBlueprint which is an abstract class extended to DeleteBlueprint, AddBlueprint, and EditBlueprint. Lastly, there are 2 entity classes, Blueprint and RawBlueprint.

Revised Software Architecture Model:



User Interface Package:

Class Name	Description
RawBlueprintSI	<p>This is the system interface of the admin to the system whenever he or she needs to provide blueprint.</p> <p><u>Attributes:</u></p> <p>private Int rawBlueprintID; private String rawBlueprintName; private Rawblueprint r;</p> <p><u>Responsibilities:</u></p> <p>public void enterRawBlueprint(int rawBlueprintID, String rawBlueprintName) public void givesRawBlueprint(RawBlueprint r)</p>
UserViewsUI	<p>This is the user interface of the user to the system whenever he or she needs to view blueprint.</p> <p><u>Attributes:</u></p> <p>private Blueprint b; private String blueprintName;</p> <p><u>Responsibilities:</u></p> <p>public void clickBlueprintToView(Blueprint b) // user clicks blueprint to view public void enterBlueprintToSearch(String blueprintName) // user searches blueprint to view public Blueprint getBlueprint(Blueprint b) // system gets the information given by the user public Blueprint getBlueprint(String blueprintName) // system gets the information given by the user</p>
DeveloperViewsUI	<p>This is the user interface of the developer whenever he or she needs to view blueprint.</p> <p><u>Attributes:</u></p> <p>private Blueprint b; private String blueprintName;</p> <p><u>Responsibilities:</u></p> <p>public void clickBlueprintToView(Blueprint b) // user clicks blueprint to view public void enterBlueprintToSearch(String blueprintName) // user searches blueprint to view public Blueprint getBlueprint(Blueprint b) // system gets the information given by the developer public Blueprint getBlueprint(String blueprintName) // system gets the information given by the developer</p>

DeveloperMaintainsSI	<p>This is the system interface of the developer whenever he or she needs to maintain blueprint.</p> <p><u>Attributes:</u></p> <p><u>Responsibilities:</u></p> <pre>private Blueprint b; private Int blueprintID; private String blueprintName; private String blueprintUpdatedata; private String blueprintdescription; public void enterMaintainBlueprintData(int blueprintID, String blueprintName, String blueprintUpdate, String blueprintDescription) public void submitBlueprintData(Blueprint b)</pre>
DigitalBlueprintSI	<p>This is the system interface of the developer whenever he or she needs to generate digital blueprint</p> <p><u>Attributes:</u></p> <pre>private RawBlueprint r; private Blueprint b; private String blueprintUpdatedata; private String blueprintdescription;</pre> <p><u>Responsibilities:</u></p> <pre>public void enterDigitalBlueprintData(RawBlueprint r, String blueprintUpdate, String blueprintDescription) public void generatesDigitalBlueprint(Blueprint b)</pre>

Controllers Package:

Controller Name	Description
GiveRawBlueprint	This is the control that gives the raw blueprint to be generated. <u>Attributes:</u> private RawBlueprint r; <u>Responsibilities:</u> public void GiveRawBlueprint(RawBlueprint r)
ViewBlueprint	This is the control that views the blueprints in the system. It extends SearchBlueprint <u>Attributes:</u> private Blueprint b; <u>Responsibilities:</u> public Blueprint ViewBlueprint(Blueprint b)
SearchBlueprint (extends ViewBlueprint)	This is the control that searches the blueprints in the system. It is extended by ViewBlueprint. <u>Attributes:</u> private Blueprint b; private String blueprintName; <u>Responsibilities:</u> public Blueprint SearchBlueprint(String blueprintName)
MaintainBlueprint (abstract)	This is the control that maintains the blueprints in the system. It is considered an abstract class. <u>Attributes:</u> private Blueprint b; private String blueprintName; private String blueprintUpdatedata; private String blueprintdescription;
AddBlueprint (extends MaintainBlueprint)	This is the control that adds a blueprint to the system. It extends MaintainBlueprintController. <u>Attributes:</u> private Blueprint b; <u>Responsibilities:</u> public void AddBlueprint(Blueprint b)
DeleteBlueprint (extends MaintainBlueprint)	This is the control that deletes a blueprint in the system. It extends MaintainBlueprintController. <u>Attributes:</u> private Blueprint b; <u>Responsibilities:</u> public void DeleteBlueprint(Blueprint b)

EditBlueprint (extends MaintainBlueprint)	<p>This is the control that edits a blueprint in the system. It extends MaintainBlueprintController.</p> <p><u>Attributes:</u></p> <pre>private Blueprint b; private String blueprintName; private String blueprintUpdatedata; private String blueprintdescription;</pre> <p><u>Responsibilities:</u></p> <pre>public void EditBlueprint(Blueprint b,String blueprintName, String blueprintUpdatedata, String blueprintDescription)</pre>
GenerateBlueprint	<p>This is the control that generates the digital version of the given raw blueprint.</p> <p><u>Attributes:</u></p> <pre>private Blueprint b;</pre> <p><u>Responsibilities:</u></p> <pre>public Blueprint GenerateBlueprint(Blueprint b)</pre>

Data Access Objects Packages:

DAO Name	Description
BuildingDAO	<p>This is the entity building which contains the data about the blueprint of the building.</p> <p><u>Attributes:</u></p> <p>private char BuildID; private String BuildName;</p> <p><u>Methods:</u></p> <p>private void addBlueprint(Blueprint b); private void updateBlueprint(Blueprint b); private void deleteBlueprint(Blueprint b);</p>
FloorDAO	<p>This is the entity floor which contains the data about the floor of the building.</p> <p><u>Attributes:</u></p> <p>private char BuildID; private char FloorID; private int FloorNo; private url floorImageLink;</p>

Transfer Objects Package:

Class Name	Description
Building	<p>This is the entity class Building, which contains the data about the Building..</p> <p><u>Attributes:</u></p> <pre>private char buildID; private String buildName;</pre> <p><u>Methods:</u></p> <pre>private void setBuildID(char bid); private void setBuildName(String bName); ... private int getBuildingID(); private String getBuildingName();</pre>
BuildingList	<p>This is the entity class BuildingList, which contains the data about the BuildingList</p> <p><u>Attributes:</u></p> <pre>private ArrayList<String> blueprints</pre> <p><u>Methods:</u></p> <pre>private void addBuildingToList(int bid, String bName); ... private int getCount(); private ArrayList<String> getList();</pre>
Floor	<p>This is the entity class Floor, which contains the data about the Floor..</p> <p><u>Attributes:</u></p> <pre>private char floorID; private String floorName; private int FloorNo; private url floorImageLink;</pre> <p><u>Methods:</u></p> <pre>private void setFloorID(char bid); private void setFloorName(String bName); private void setFloorNo(int x); private set floorImageLink(url x); ...</pre>

	<pre>private int getFloorID(); private String getFloorName(); private url getFloorImageLink();</pre>
BuildingList	<p>This is the entity class BuildingList, which contains the data about the BuildingList</p> <p><u>Attributes:</u></p> <pre>private ArrayList<String> blueprints</pre> <p><u>Methods:</u></p> <pre>private void addBuildingToList(int bid, String bName); ... private int getCount(); private ArrayList<String> getList();</pre>

Data Sources Package:

File Name or Database Name	Description
BlueprintDatabase	This is the database that contains all the blueprint records for the MaroonPrint system.