

# **MaroonPrint**

## Software Architectural Design

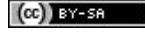
Submitted to:

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  
1<sup>st</sup> 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.

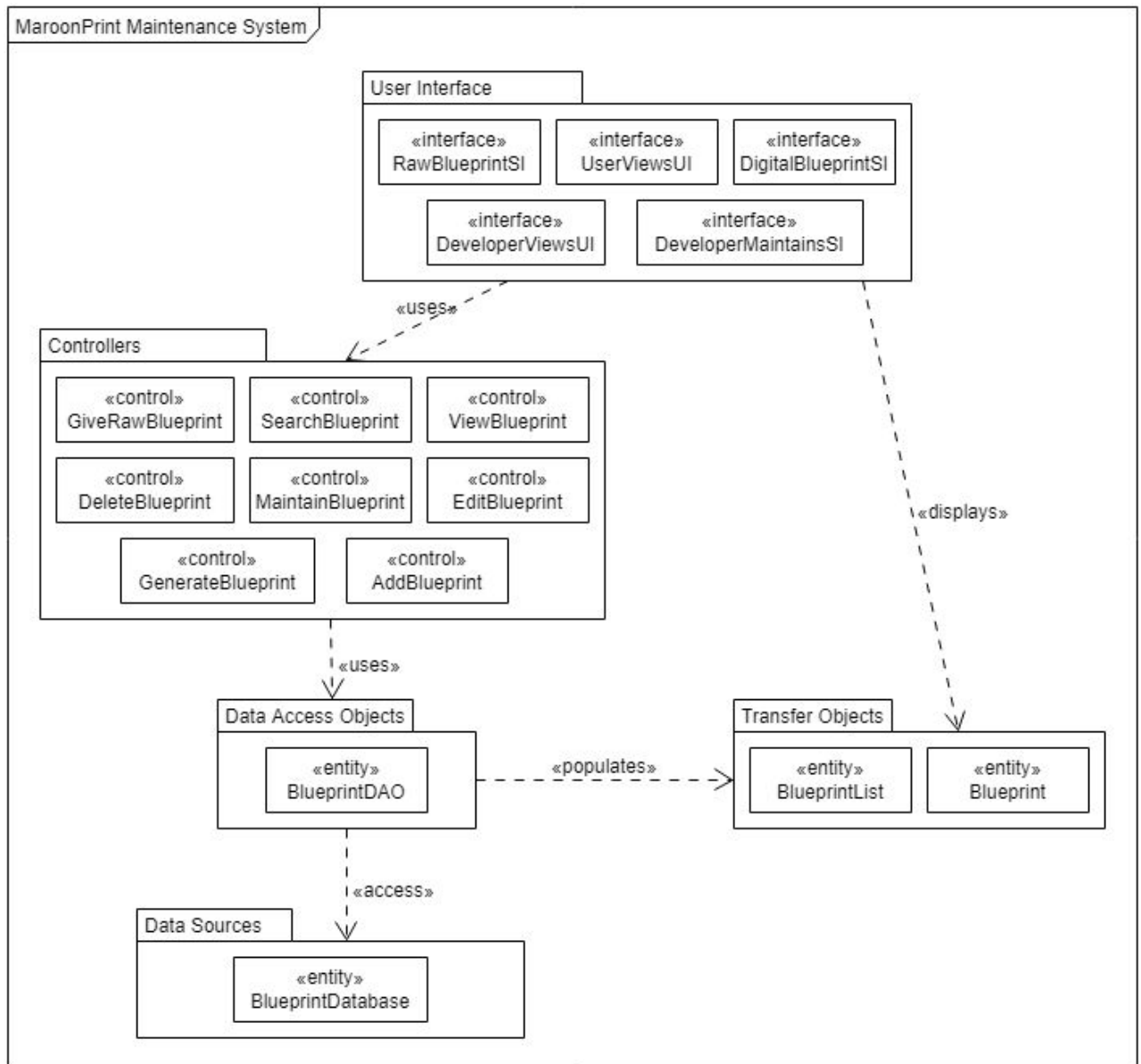
***Revision Control:***

<b><i>Revision Date</i></b>	<b><i>Person Responsible</i></b>	<b><i>Version Number</i></b>	<b><i>Contribution/Modification</i></b>
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.

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

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
BlueprintDAO	<p>This is the entity blueprint which contains the data about the blueprint.</p> <p><u>Attributes:</u></p> <p>private int blueprintID; private String blueprintName; private String blueprintUpdate; // <i>provides the last update date of the blueprint</i> private String blueprintDescription;</p> <p><u>Methods:</u></p> <p>private void addBlueprint(Blueprint b); private void updateBlueprint(Blueprint b); private void deleteBlueprint(Blueprint b);</p>

### ***Transfer Objects Package:***

Class Name	Description
Blueprint	<p>This is the entity class Blueprint, which contains the data about the Blueprint..</p> <p><u>Attributes:</u></p> <pre>private int blueprintID private String blueprintName private String blueprintUpdate //provides the last update date of the blueprint private String blueprintDescription</pre> <p><u>Methods:</u></p> <pre>private void setBlueprintID(int bid); private void setBlueprintName(String bName); private void setBlueprintUpdate(String bDate); private void set BlueprintDescription(String bDescription); ... private int getBlueprintID(); private String getBlueprintName(); private String getBlueprintUpdate(); private String getBlueprintDescription();</pre>
BlueprintList	<p>This is the entity class BlueprintList, which contains the data about the BlueprintList</p> <p><u>Attributes:</u></p> <pre>private int blueprintCount private ArrayList&lt;String&gt; blueprints</pre> <p><u>Methods:</u></p> <pre>private void addBlueprintToList(int bid, String bName); ... private int getCount(); private ArrayList&lt;String&gt; 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.