

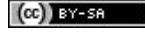
MaroonPrint Analysis Model

Submitted to:

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

Submitted by:
Lee, Kristine-Clair
Magno, Hannah Mae
Wu, Jeremy Jian 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 Group3-MaroonPrint-Analysis Model.

Purpose:

The purpose of this paper is to serve as the conceptual overview of the system that is to be developed.

Audience:

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

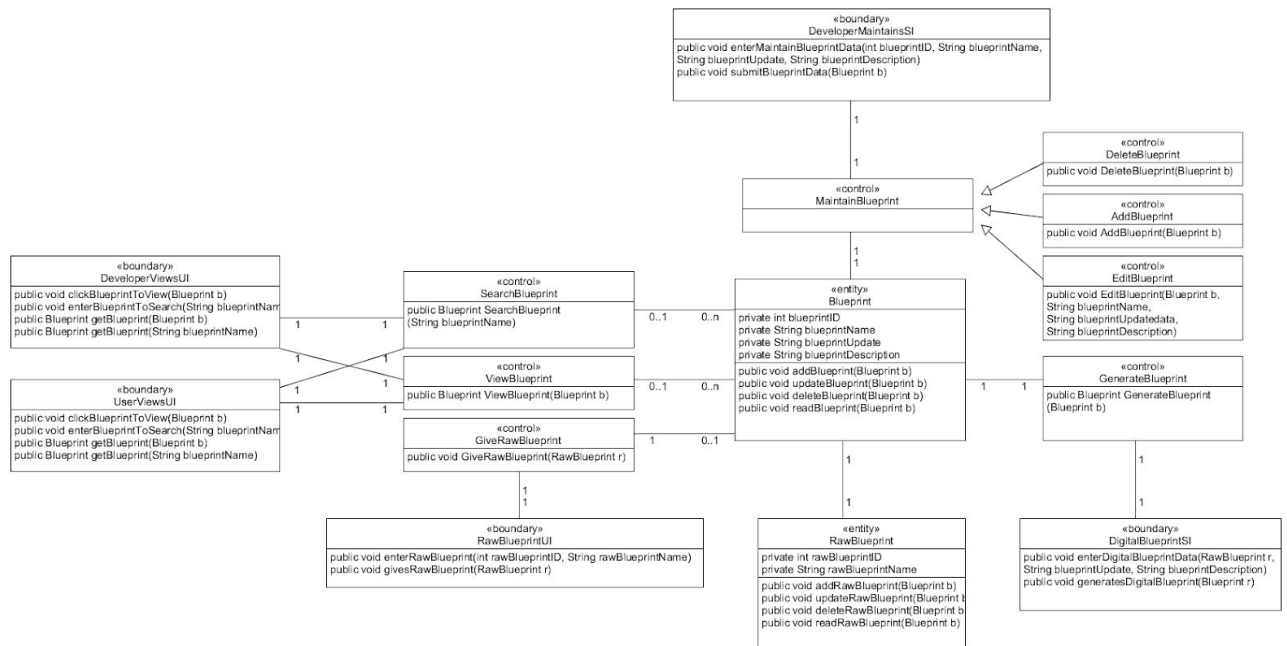
Revision Control:

<i>Revision Date</i>	<i>Person Responsible</i>	<i>Version Number</i>	<i>Modification</i>
10/01/2018	Kristine-Clair Lee	1.0	Initial Document; added entity classes and sequence diagrams for use case 1.0, 2.0, and 2.1
10/01/2018	Wu Jeremy Jin Qian	1.1	added control classes and sequence diagram for use case 3.3 and 4.0.
10/01/2018	Hannah Mae Magno	1.2	added boundary classes and sequence diagram for use case 3.0, 3.1, and 3.2; added the diagram for analysis model

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.

Analysis Model:



Boundary Classes:

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>Responsibilities:</u></p> <pre>public void enterRawBlueprint(int rawBlueprintID, String rawBlueprintName) public void givesRawBlueprint(RawBlueprint r)</pre>
UserViewsUI	<p>This is the user interface of the user to the system whenever he or she needs to view blueprint.</p> <p><u>Responsibilities:</u></p> <pre>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</pre>
DeveloperViewsUI	<p>This is the user interface of the developer whenever he or she needs to view blueprint.</p> <p><u>Responsibilities:</u></p> <pre>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</pre>
DeveloperMaintainsSI	<p>This is the system interface of the developer whenever he or she needs to maintain blueprint.</p> <p><u>Responsibilities:</u></p> <pre>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>Responsibilities:</u></p> <pre>public void enterDigitalBlueprintData(RawBlueprint r, String blueprintUpdate, String blueprintDescription) public void generatesDigitalBlueprint(Blueprint r)</pre>

Control Classes:

Class Name	Description
GiveRawBlueprint	This is the control that gives the raw blueprint to be generated. <u>Responsibilities:</u> public void GiveRawBlueprint(RawBlueprint r)
ViewBlueprint	This is the control that views the blueprints in the system. It extends SearchBlueprint <u>Responsibilities:</u> public Blueprint ViewBlueprint(Blueprint b)
SearchBlueprint	This is the control that searches the blueprints in the system. It is extended by ViewBlueprint. public Blueprint SearchBlueprint(String blueprintName)
MaintainBlueprint	This is the control that maintains the blueprints in the system. It is considered an abstract class.
AddBlueprint	This is the control that adds a blueprint to the system. It extends MaintainBlueprintController. <u>Responsibilities:</u> public void AddBlueprint(Blueprint b)
DeleteBlueprint	This is the control that deletes a blueprint in the system. It extends MaintainBlueprintController. <u>Responsibilities:</u> public void DeleteBlueprint(Blueprint b)
EditBlueprint	This is the control that edits a blueprint in the system. It extends MaintainBlueprintController. <u>Responsibilities:</u> public void EditBlueprint(Blueprint b,String blueprintName, String blueprintUpdatedata, String blueprintDescription)
GenerateBlueprint	This is the control that generates the digital version of the given raw blueprint. <u>Responsibilities:</u> public Blueprint GenerateBlueprint(Blueprint b)

Entity Classes:

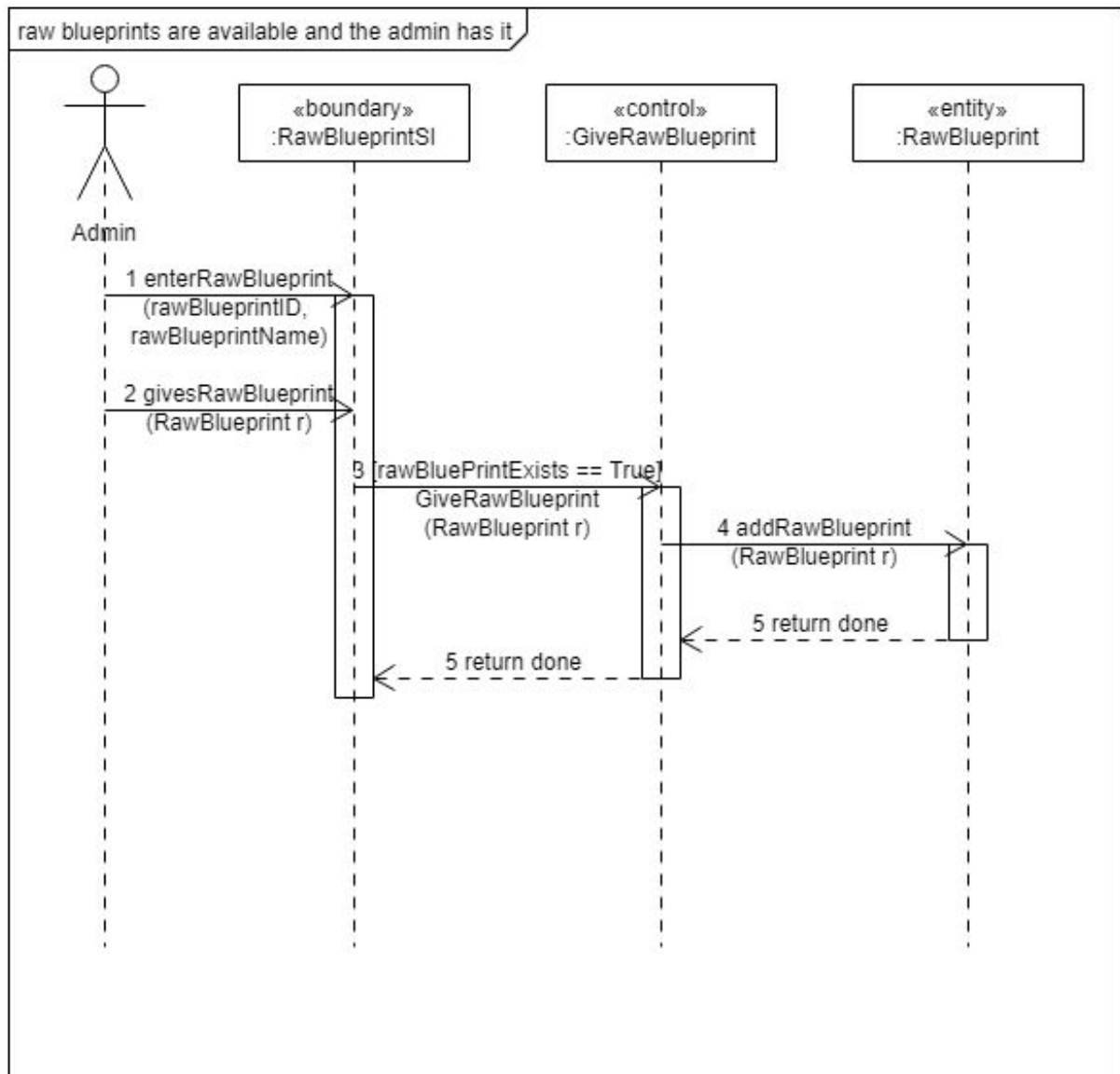
Class Name	Description
RawBlueprint	<p>This is the entity class raw blueprint, which contains the data about the raw blueprint.</p> <p>Attributes:</p> <p>private int rawBlueprintID</p> <p>private String rawBlueprintName</p>
Blueprint	<p>This is the entity class blueprint, which contains the data about the blueprint.</p> <p>Attributes:</p> <p>private int blueprintID</p> <p>private String blueprintName</p> <p>private String blueprintUpdate // provides the last update date of the blueprint</p> <p>private String blueprintDescription</p>

Behavioral Model:

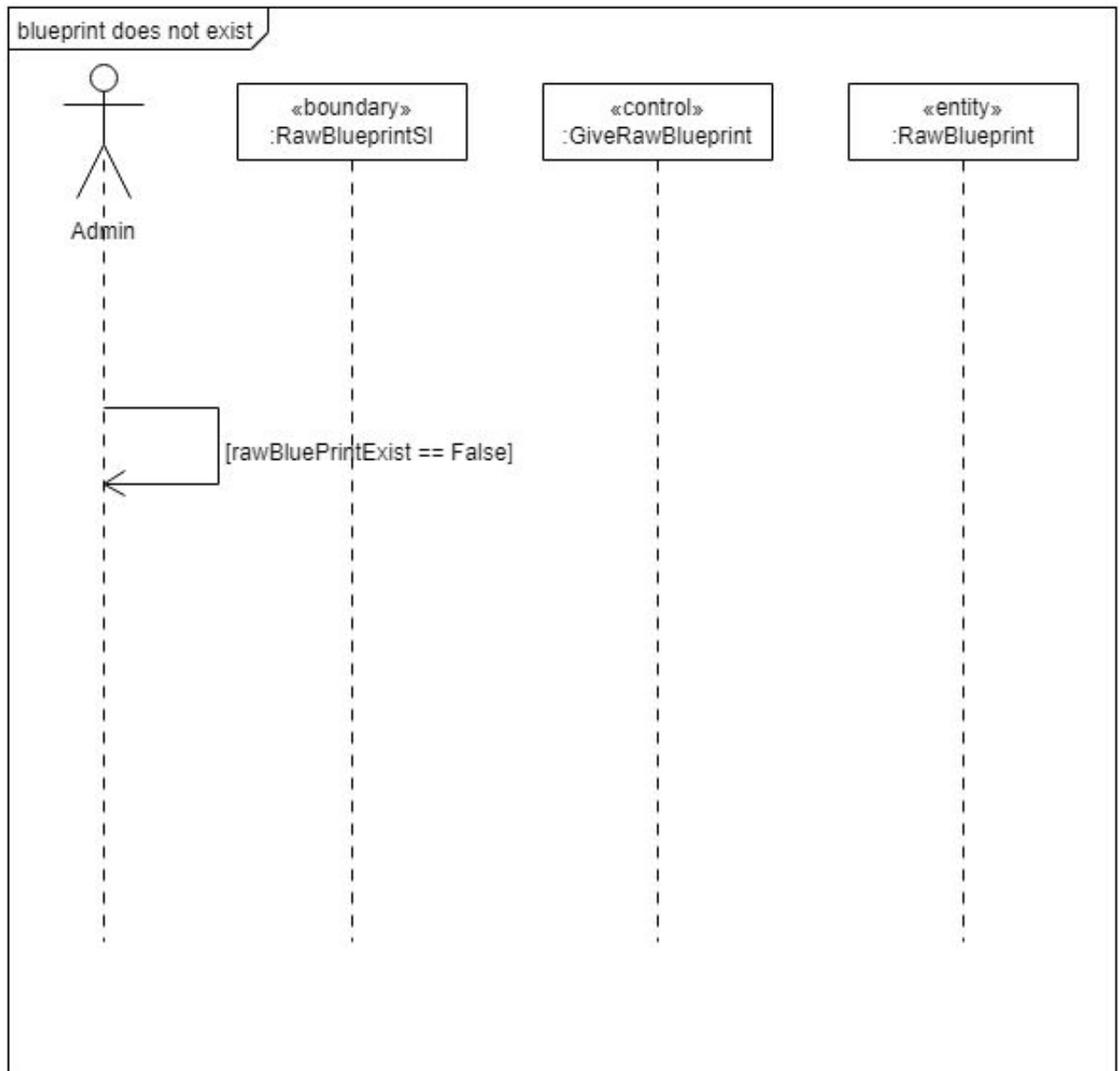
Use-Case Name: 1.0 Gives Raw Blueprints

Description: In this use case, it talks about the role of the Admin to the development and maintenance of the MaroonPrint application. The Admin's role is to provide the raw blueprints. In addition, these raw blueprints will be used to generate digital version of the blueprints for the database.

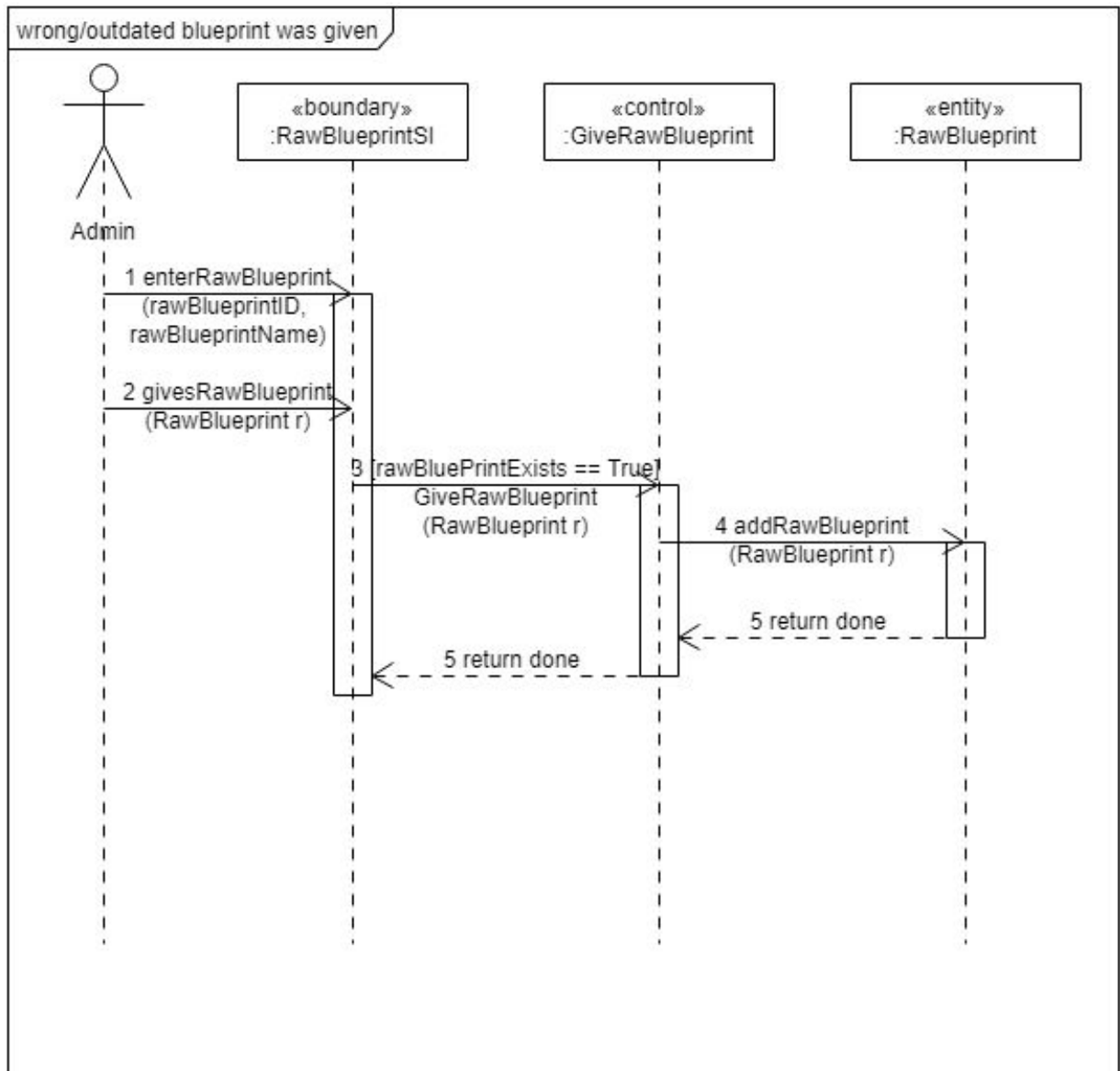
Scenario 1: Raw Blueprints are available and the Admin has it (Basic Flow)



Scenario 2: Raw blueprints are not available



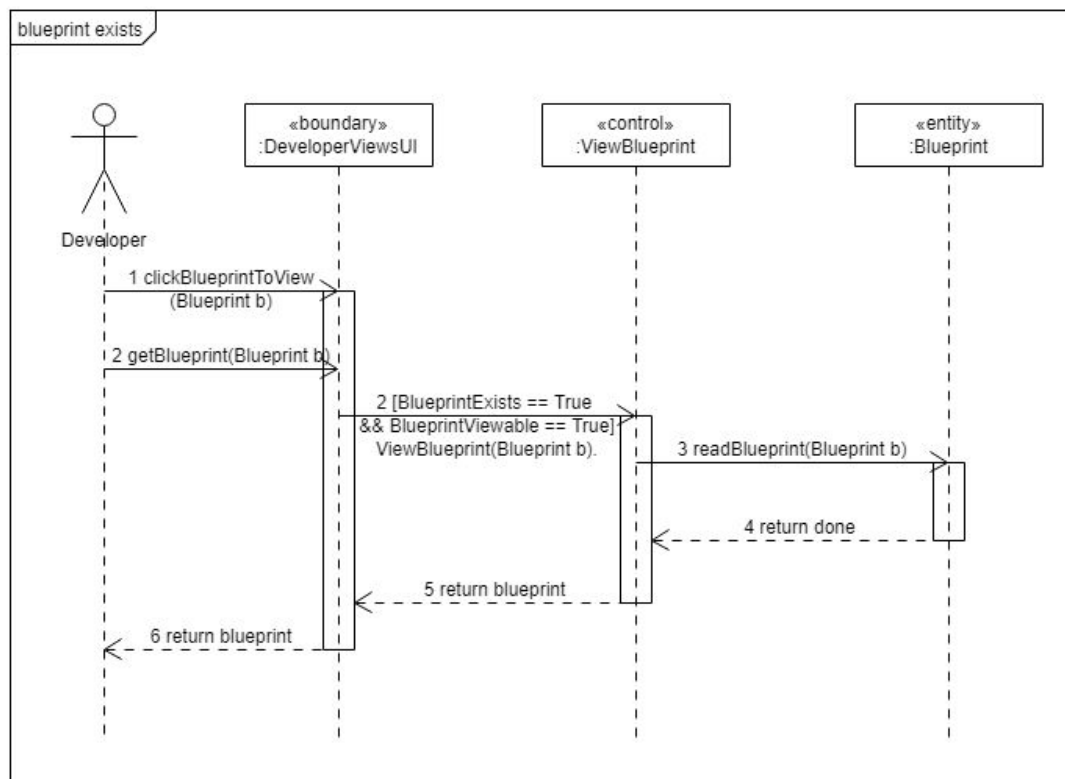
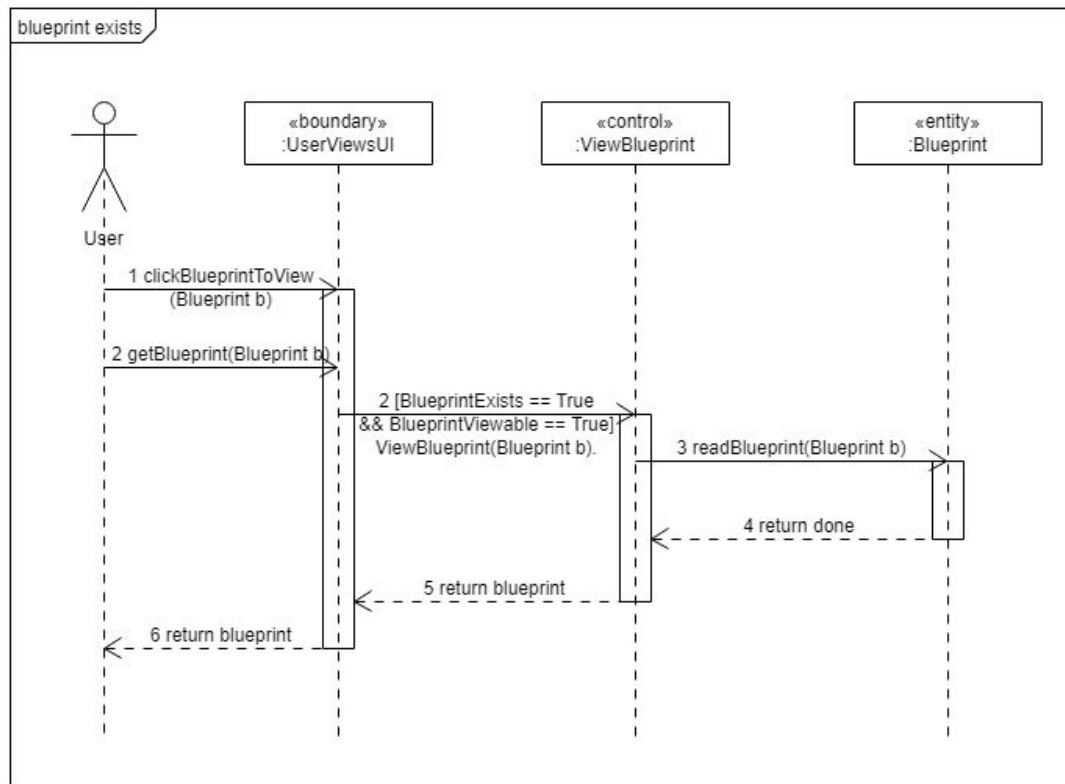
Scenario 3: Wrong/outdated blueprint was given



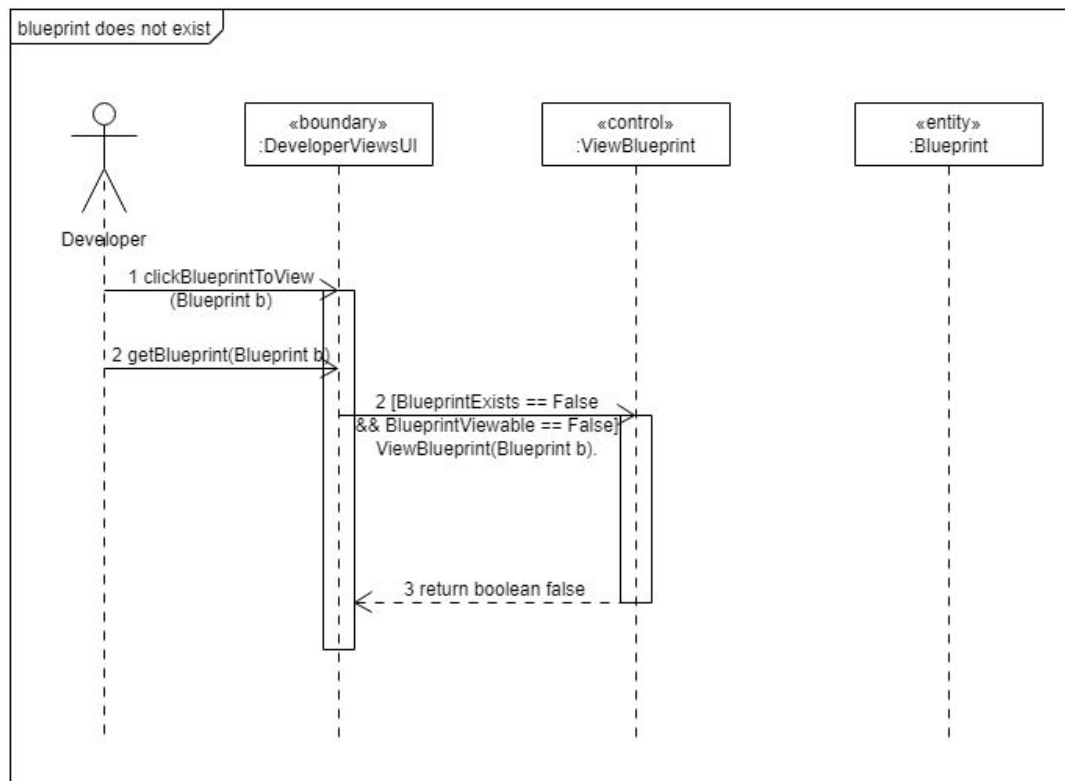
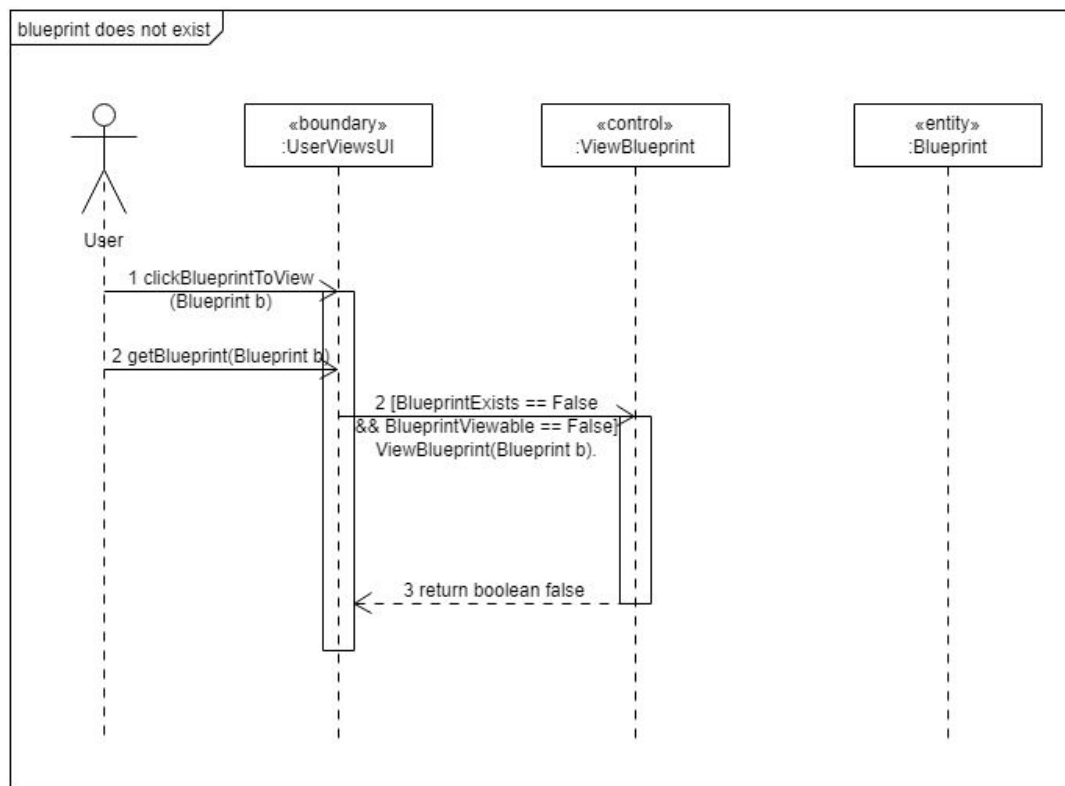
Use-Case Name: 2.0 Views Blueprints

Description: In this use case, the user and the developer gets to view blueprints that are inside the system already. In addition, the user and/or developer can search for specific building's blueprint and/or search for a specific part or area of the building.

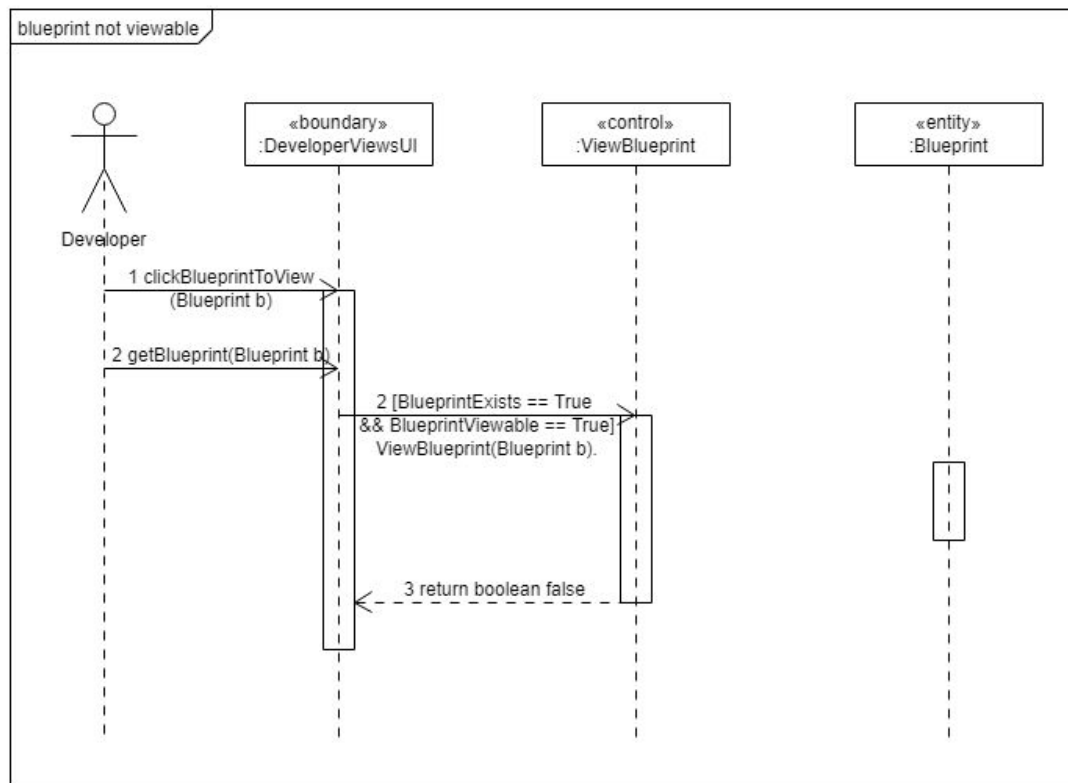
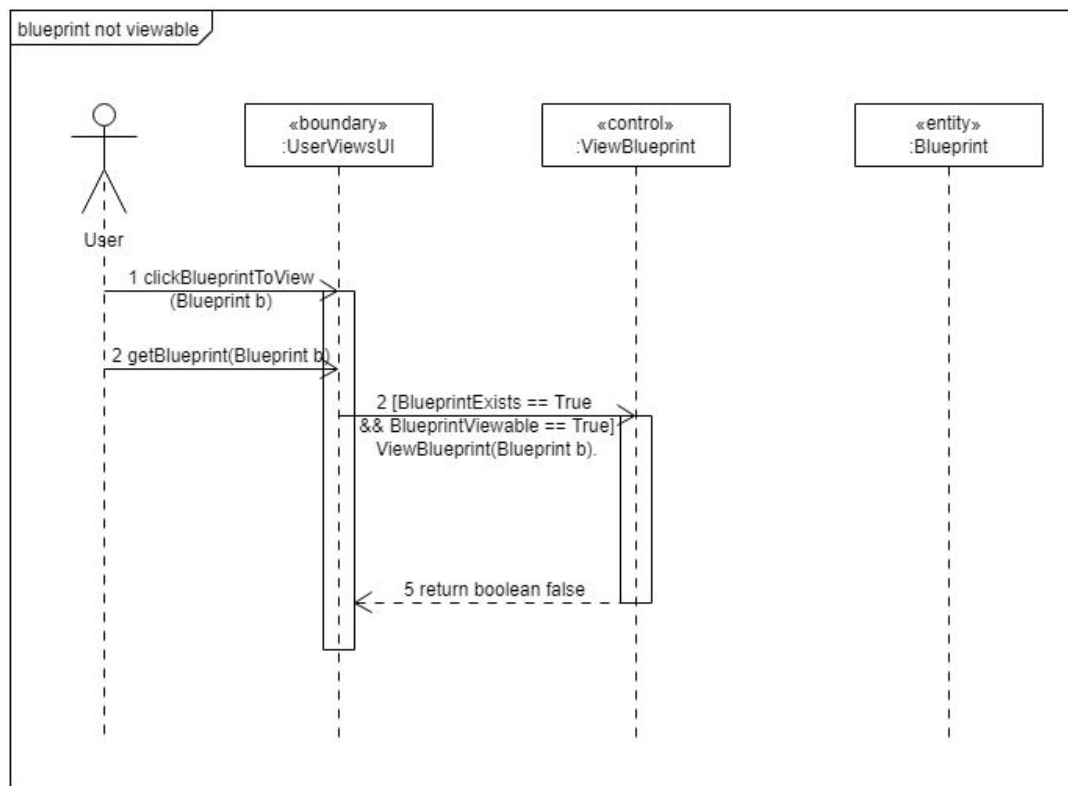
Scenario 1: Blueprint exists (Basic Flow)



Scenario 2: Blueprint does not exist



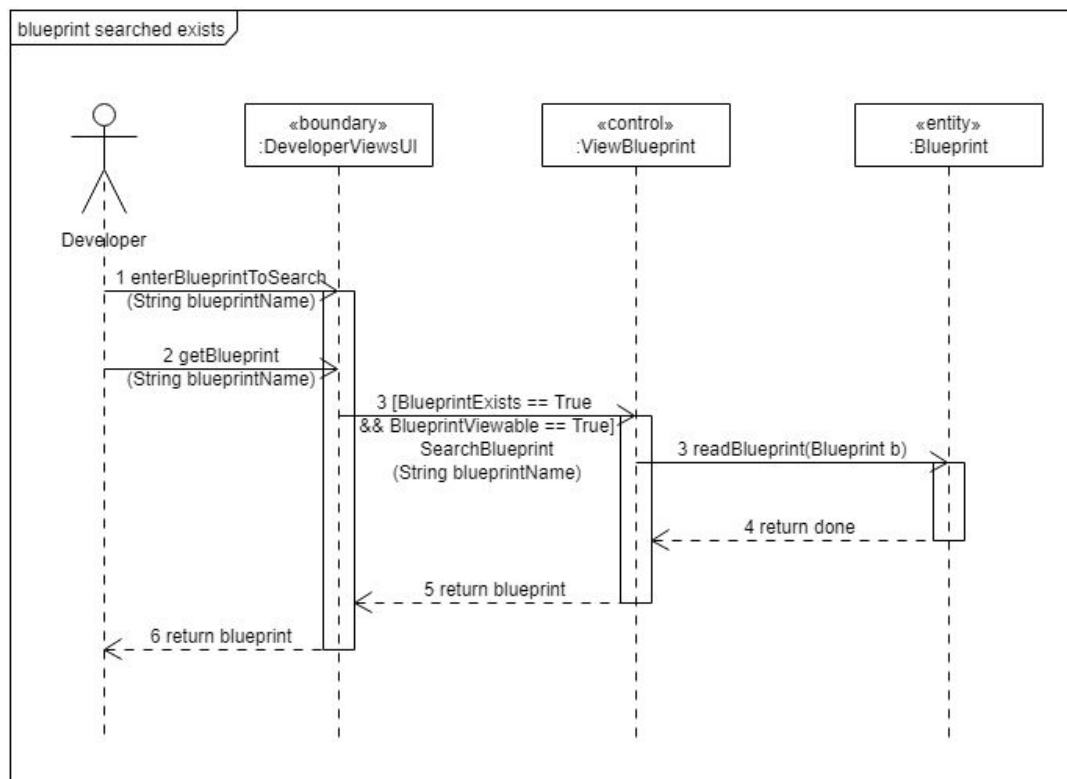
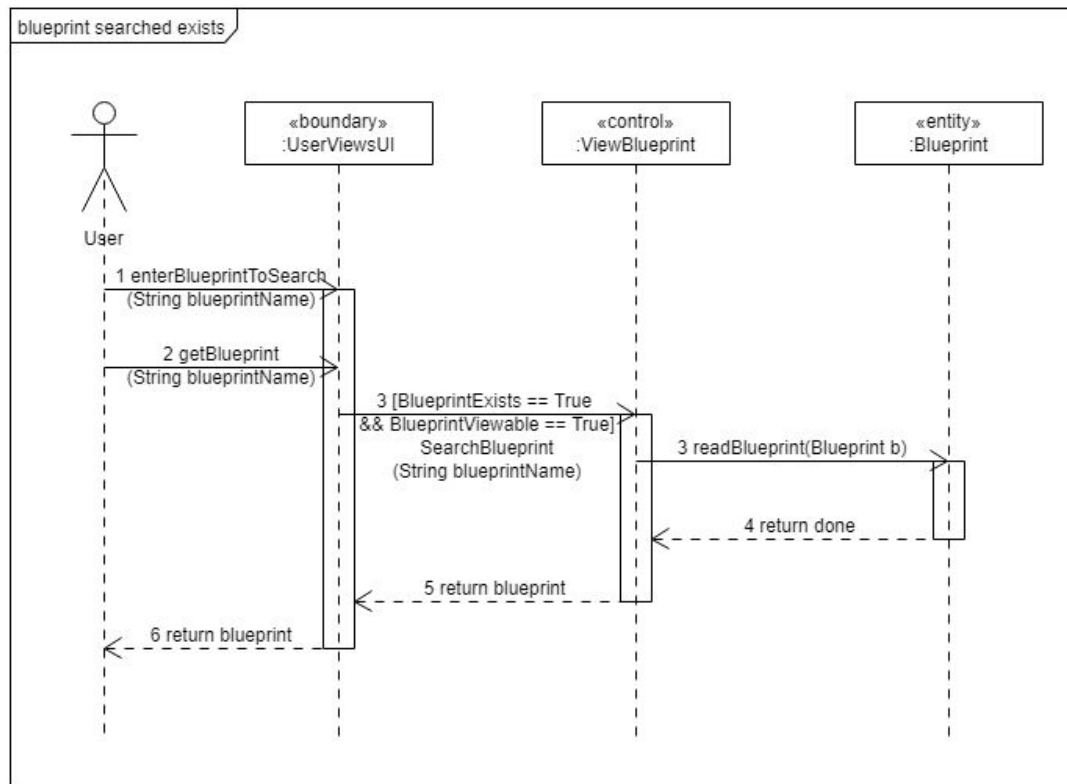
Scenario 3: Blueprint exist but is not viewable



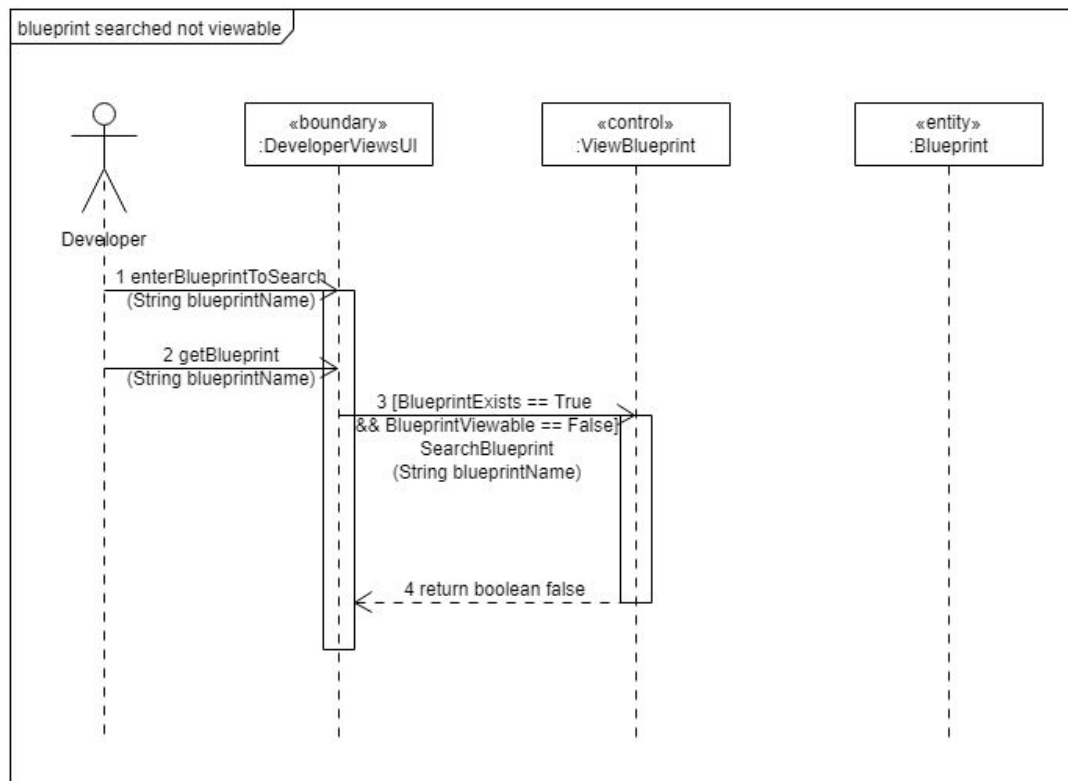
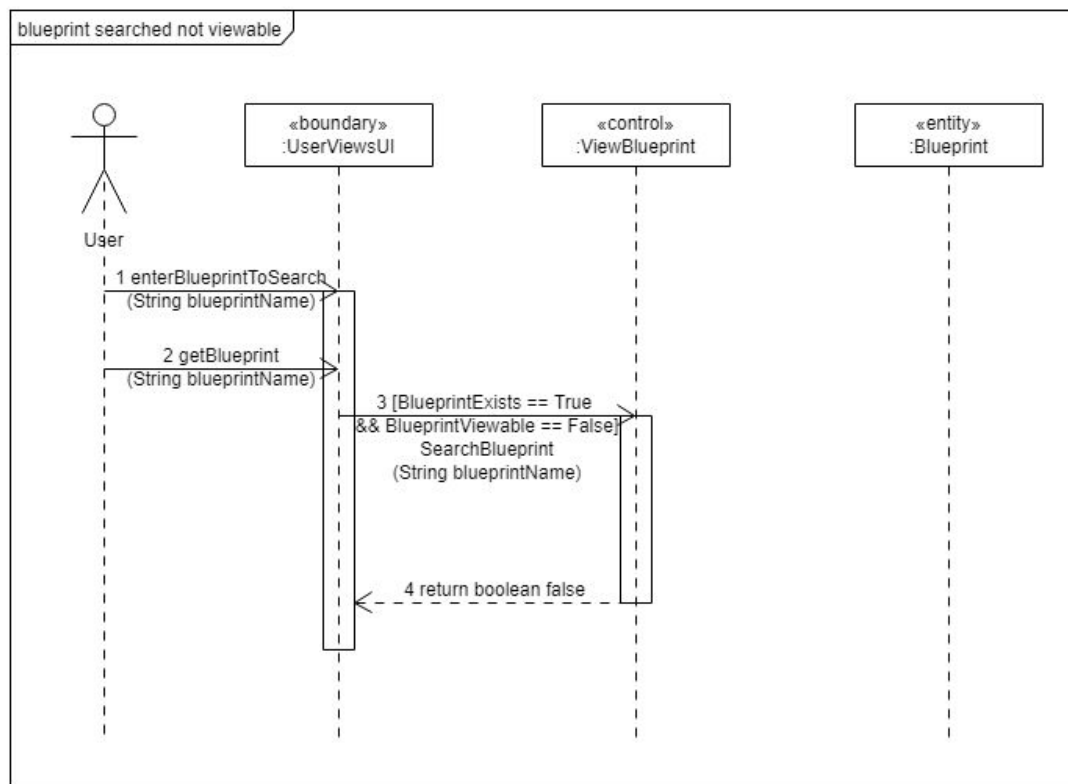
Use-Case Name: 2.1 Search Blueprint

Description: In this use case, it is the extension of the 2.0 View Blueprint use case. In this use case, both the user and the developer can search for specific building blueprints or search for specific areas of the building.

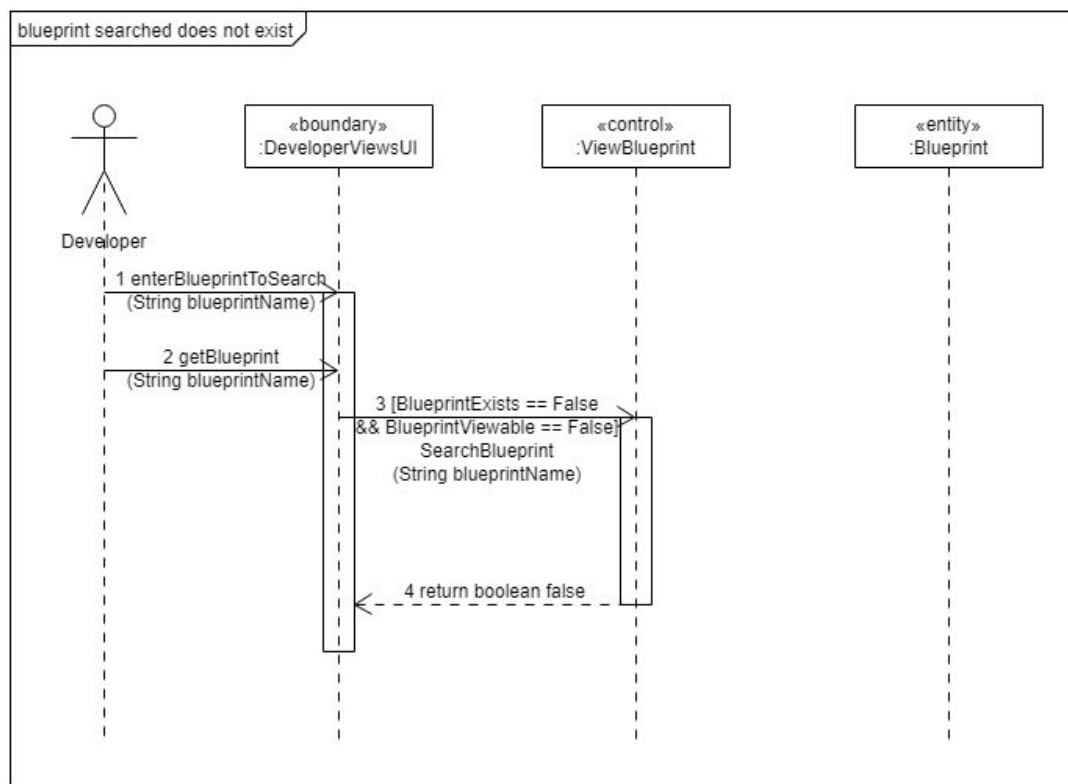
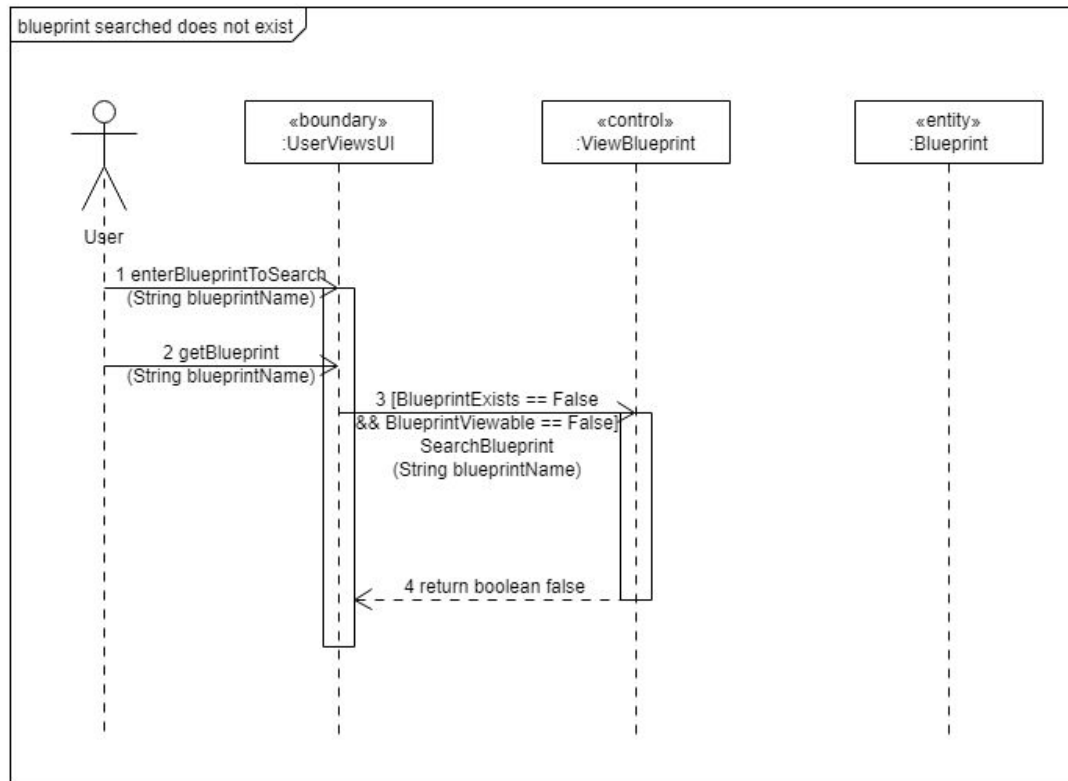
Scenario 1: Blueprint exists and is viewable (Basic Flow)



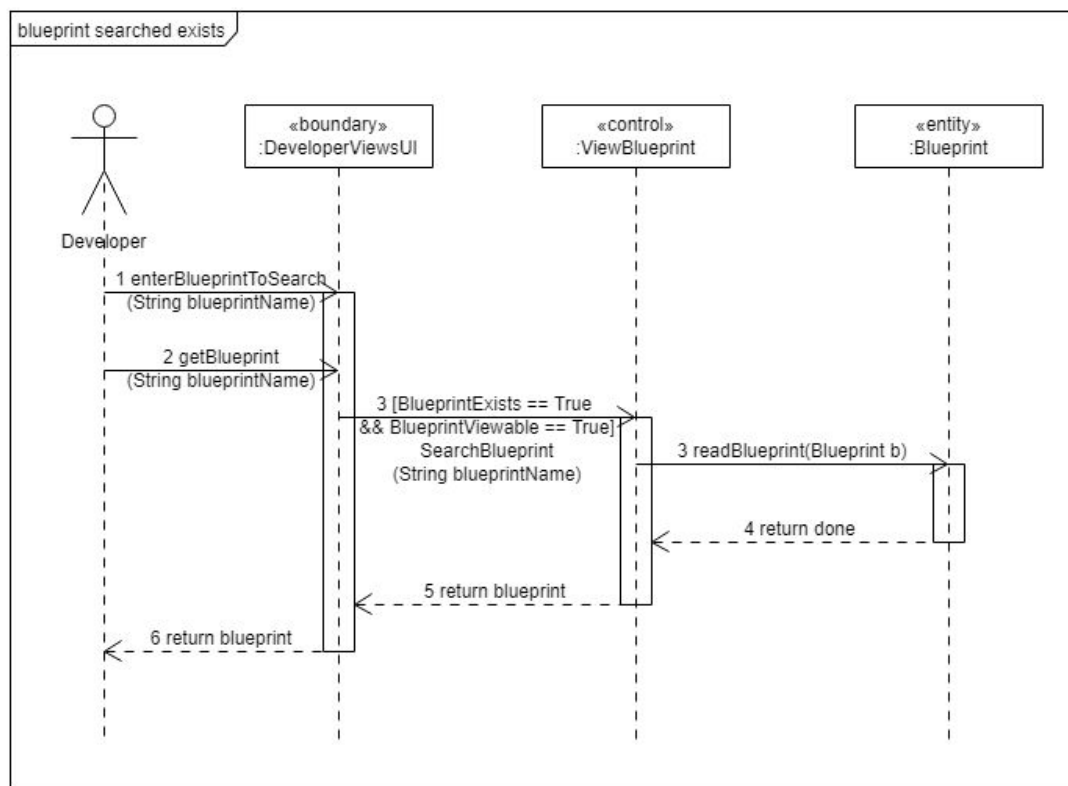
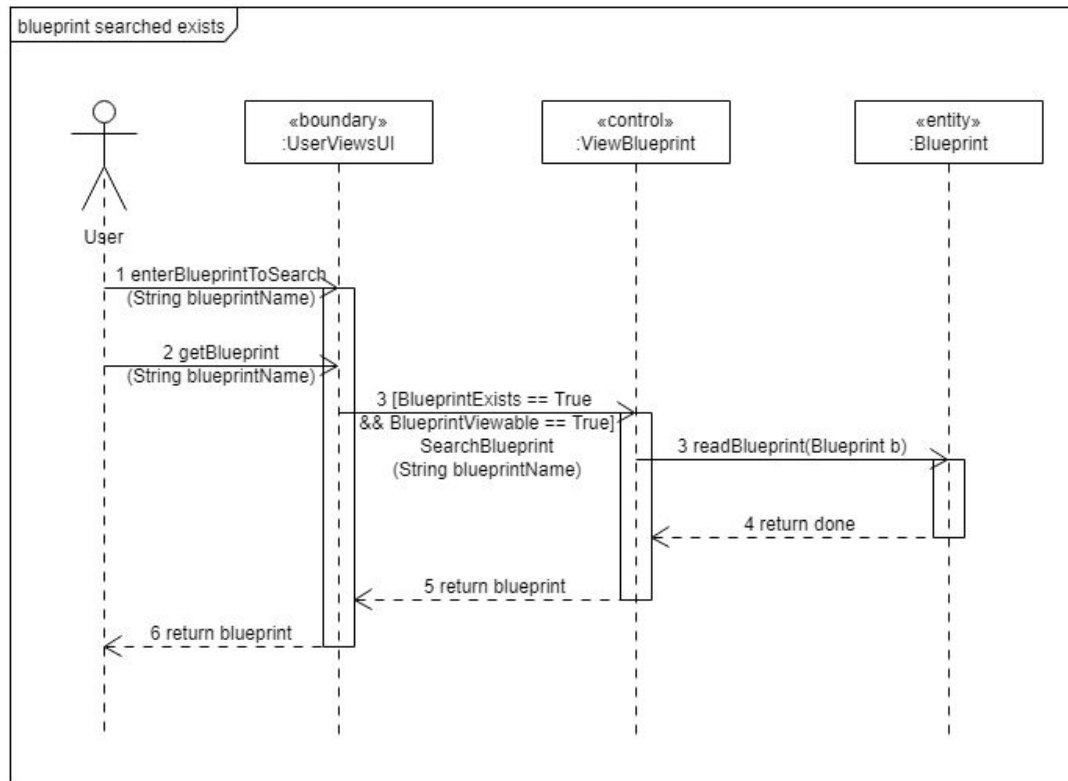
Scenario 2: Blueprint exists but is not viewable



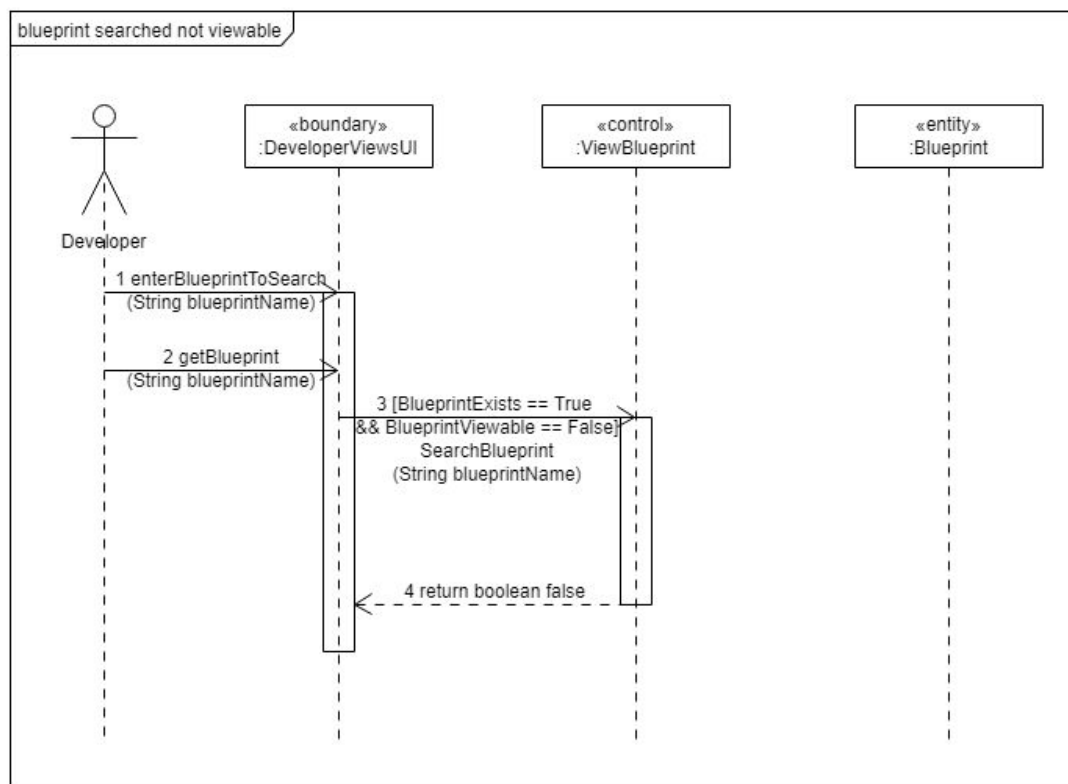
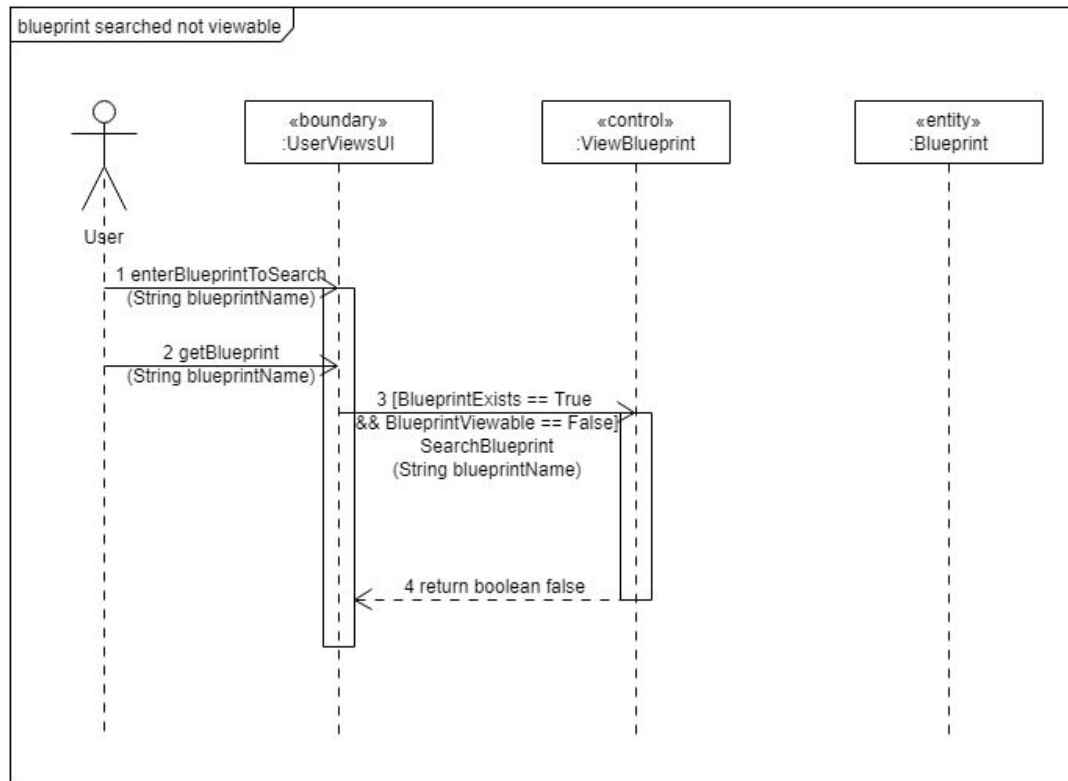
Scenario 3: Blueprint does not exist



Scenario 4: Floor blueprint exists and is viewable



Scenario 5: Floor blueprint exists but is not viewable

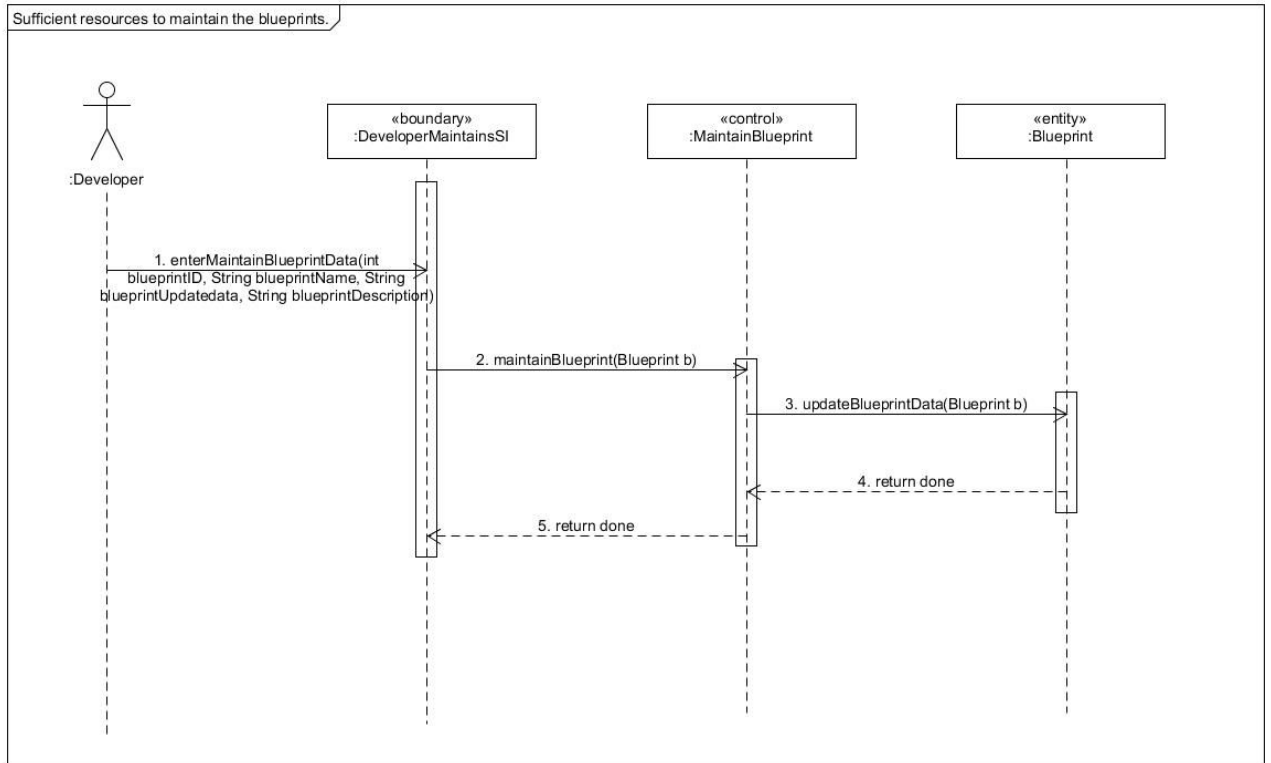


Use-Case Name: 3.0 Maintain Blueprint

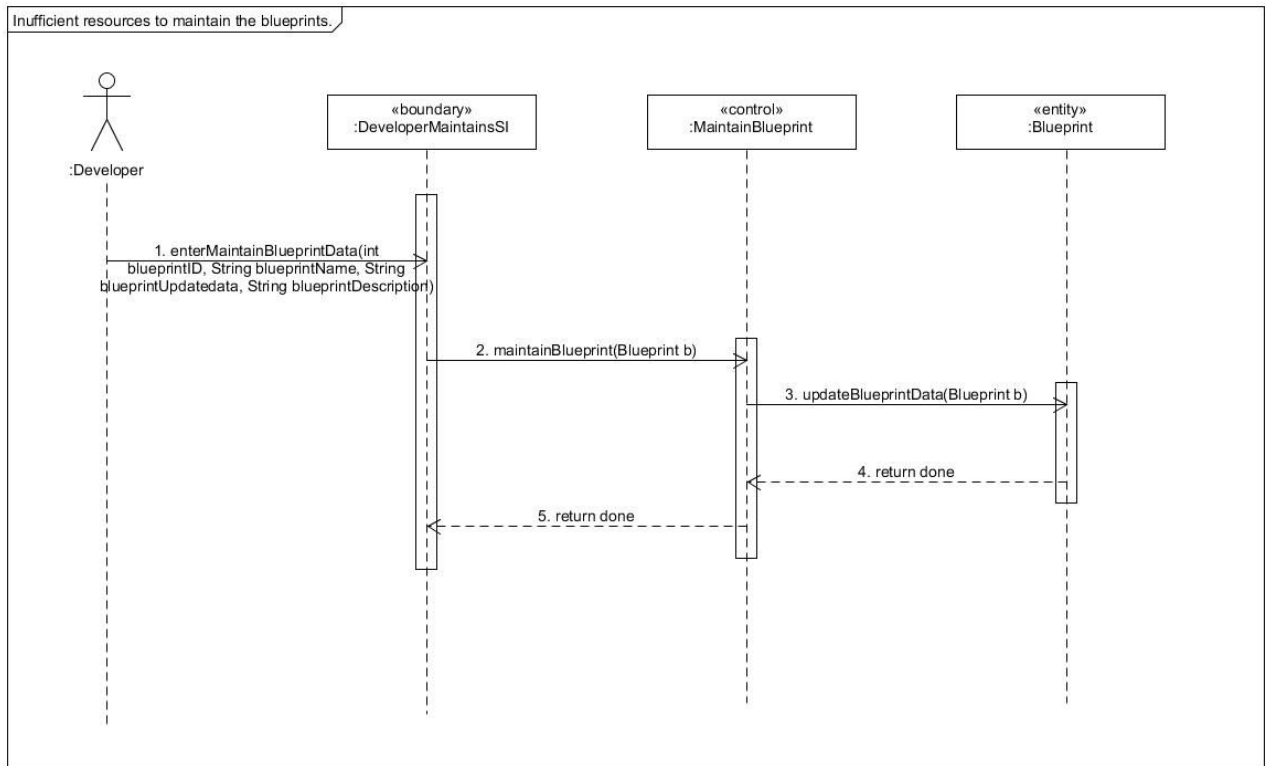
Description:

In this use case, it talks about the role of the Developer to the development and maintenance of the MaroonPrint application. The Developer's role is to accept the blueprints provided by the Admin. In addition, the Developer will maintain blueprints by adding, deleting, and editing contents to them.

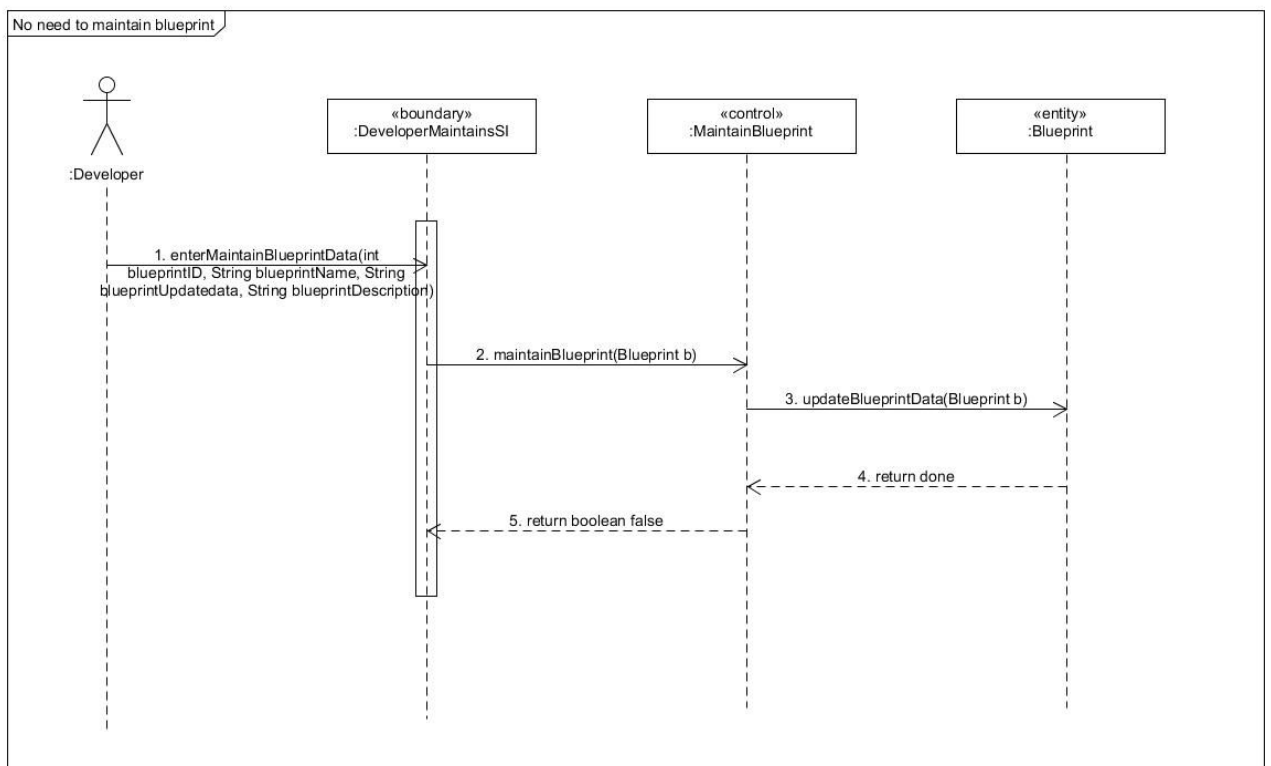
Scenario 1: Developer has sufficient resources to maintain the blueprints. (Basic Flow)



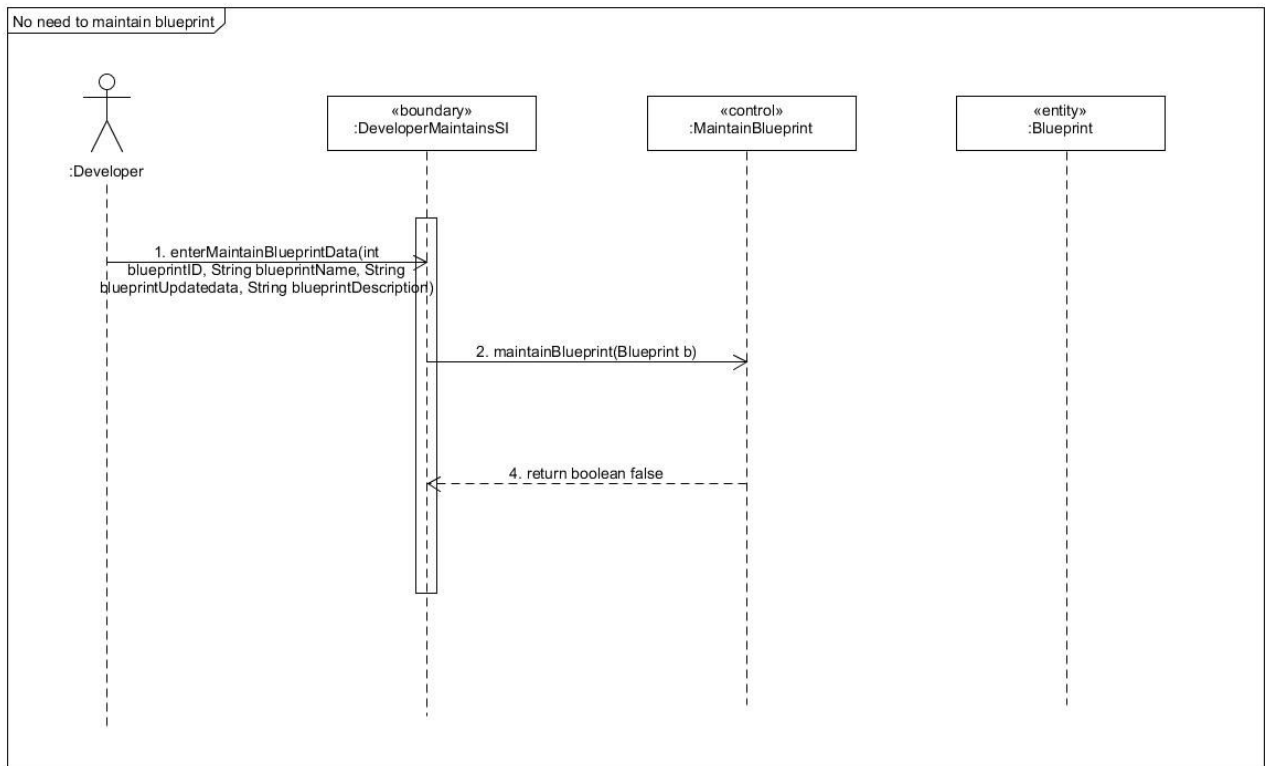
Scenario 2: Developer has insufficient resources to maintain the blueprints.



Scenario 3: No need to maintain the blueprints.



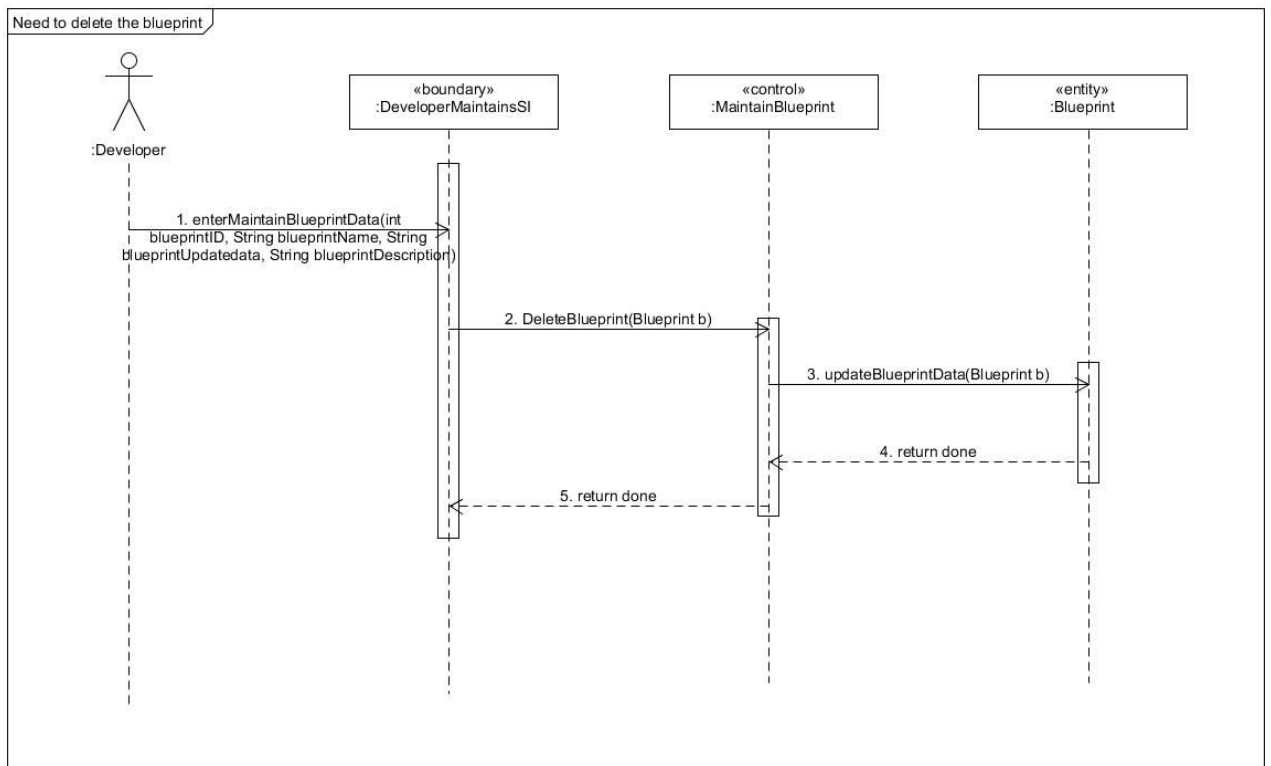
Scenario 4: No blueprint to maintain.



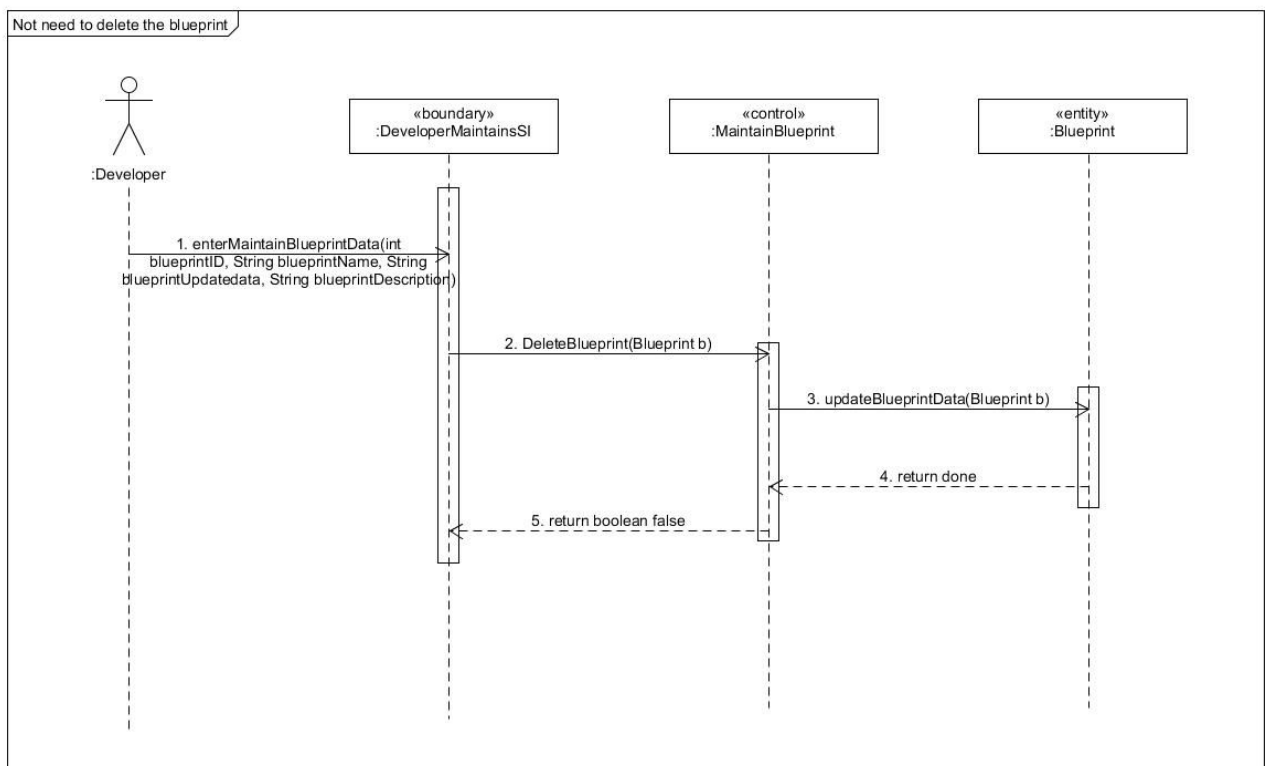
Use-Case Name: 3.1 Delete Blueprint

Description: In this use case, it talks about the role of the Developer to the development and maintenance of the MaroonPrint application. The Developer's role is to accept the blueprints provided by the Admin. In addition, the Developer can delete blueprints.

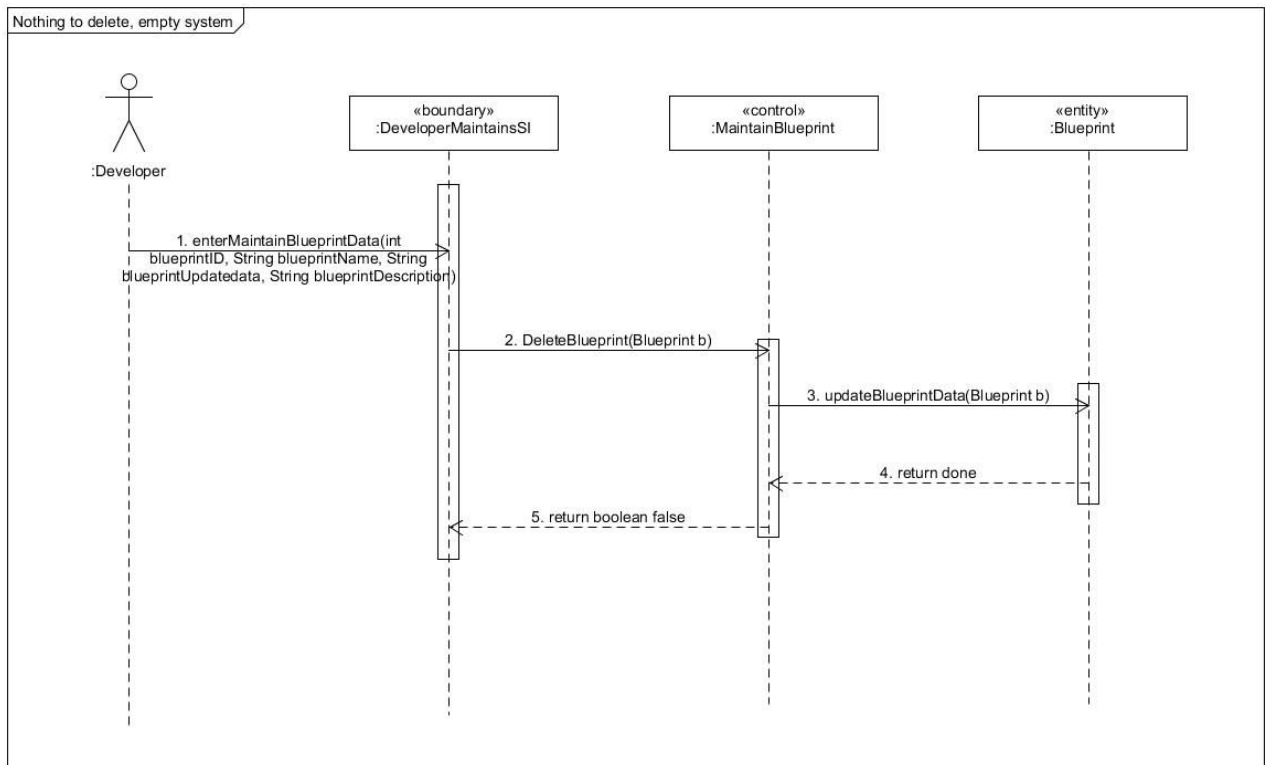
Scenario 1: There is a need to delete the blueprint. (Basic Flow)



Scenario 2: No need to delete the blueprint.



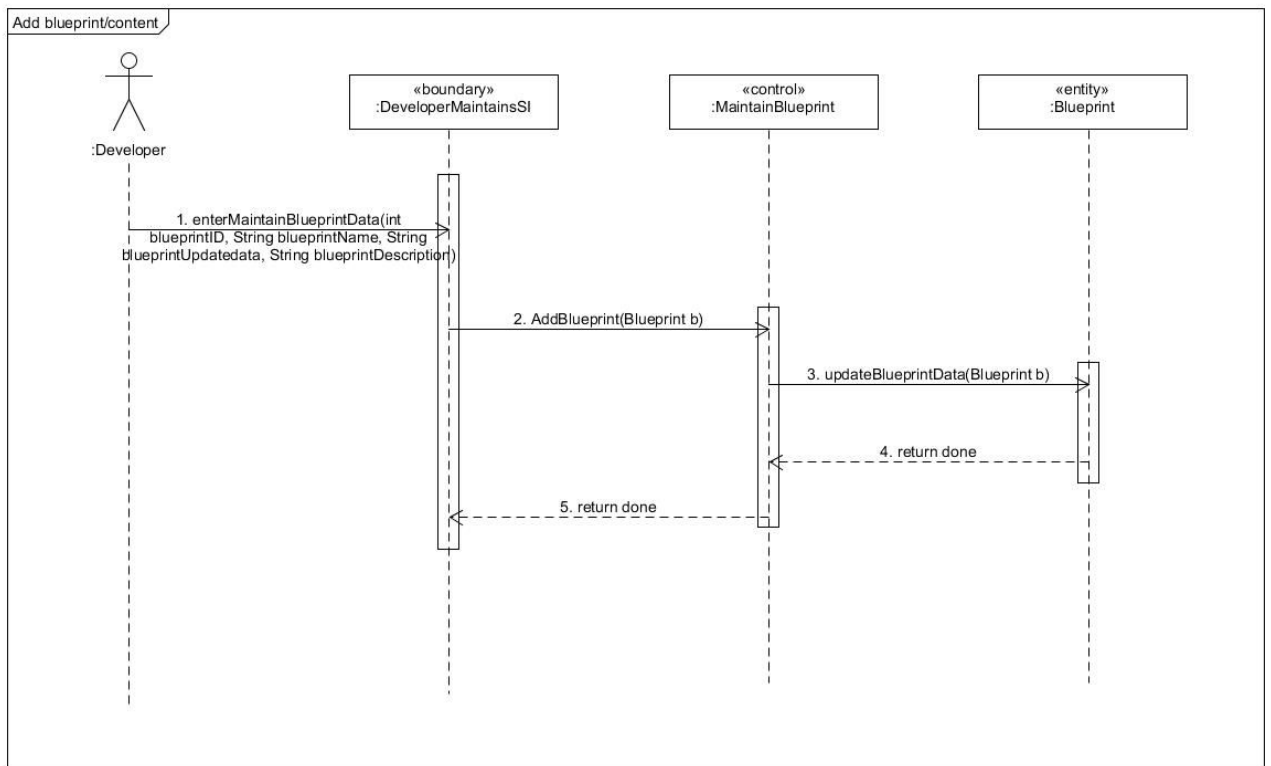
Scenario 3: Nothing to delete, empty system.



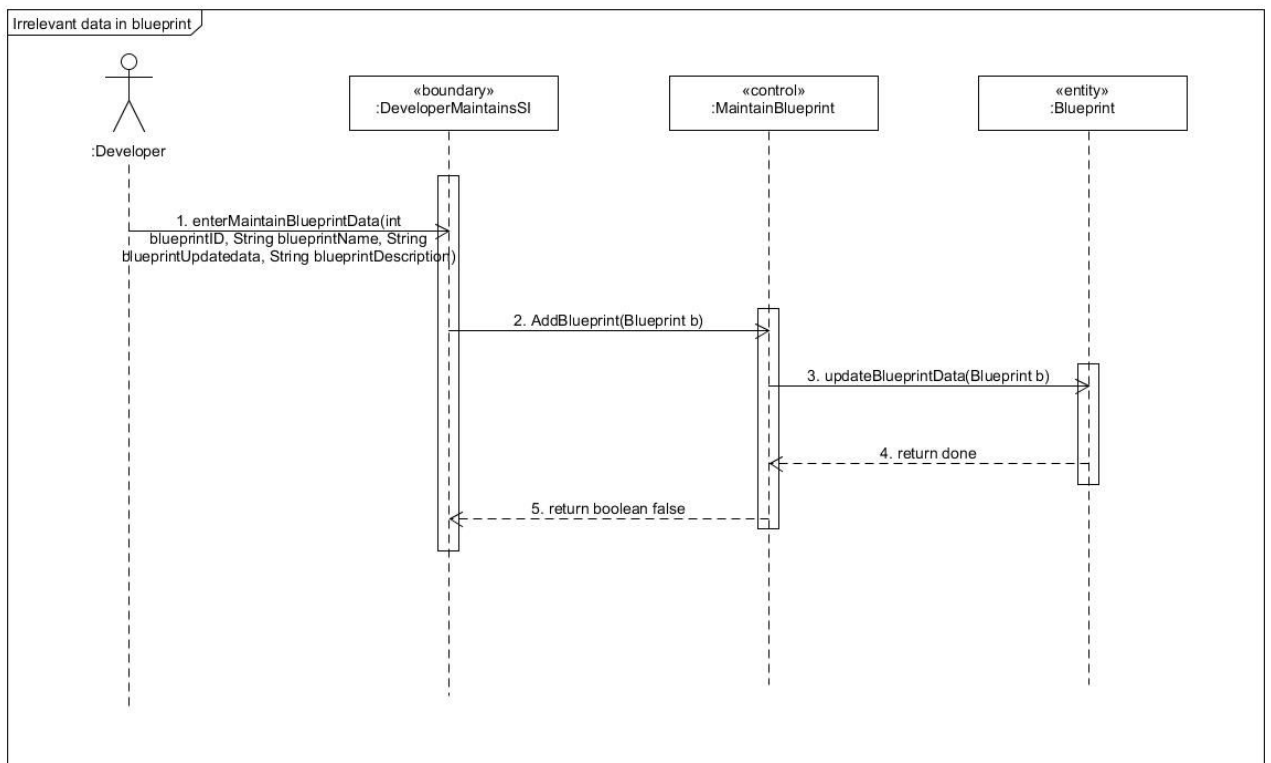
Use-Case Name: 3.2 Add blueprint

Description: In this use case, it talks about the role of the Developer to the development and maintenance of the MaroonPrint application. The Developer's role is to accept the blueprints provided by the Admin. In addition, the Developer can add blueprints.

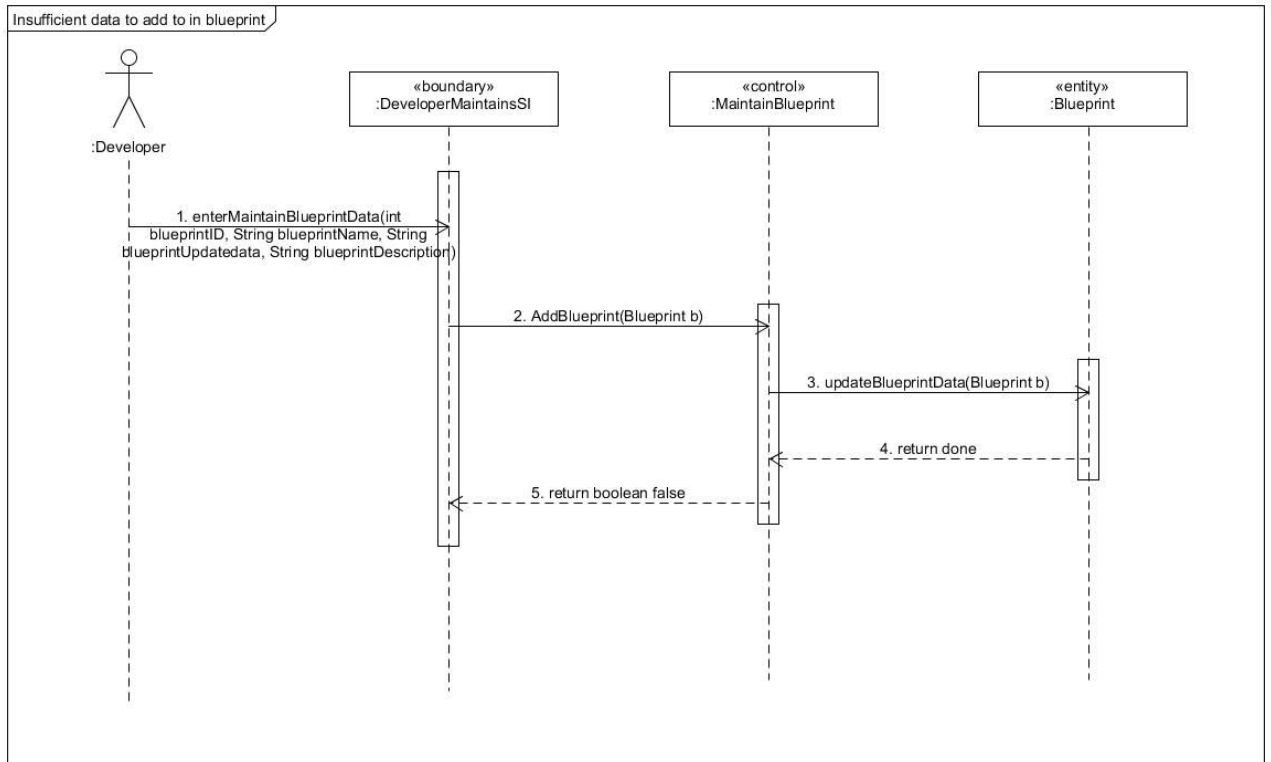
Scenario 1: Add blueprint/content. (Basic Flow)



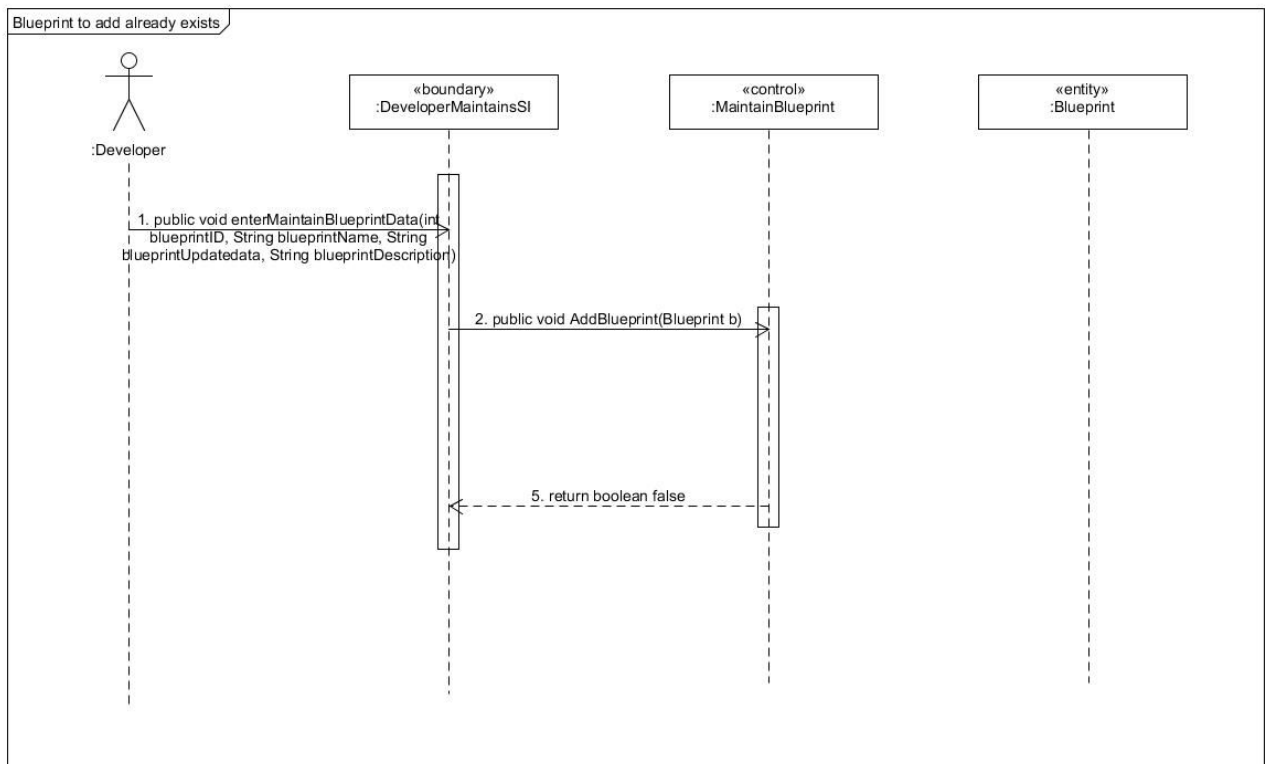
Scenario 2: Irrelevant data in blueprint



Scenario 3: Insufficient data to add to in blueprint



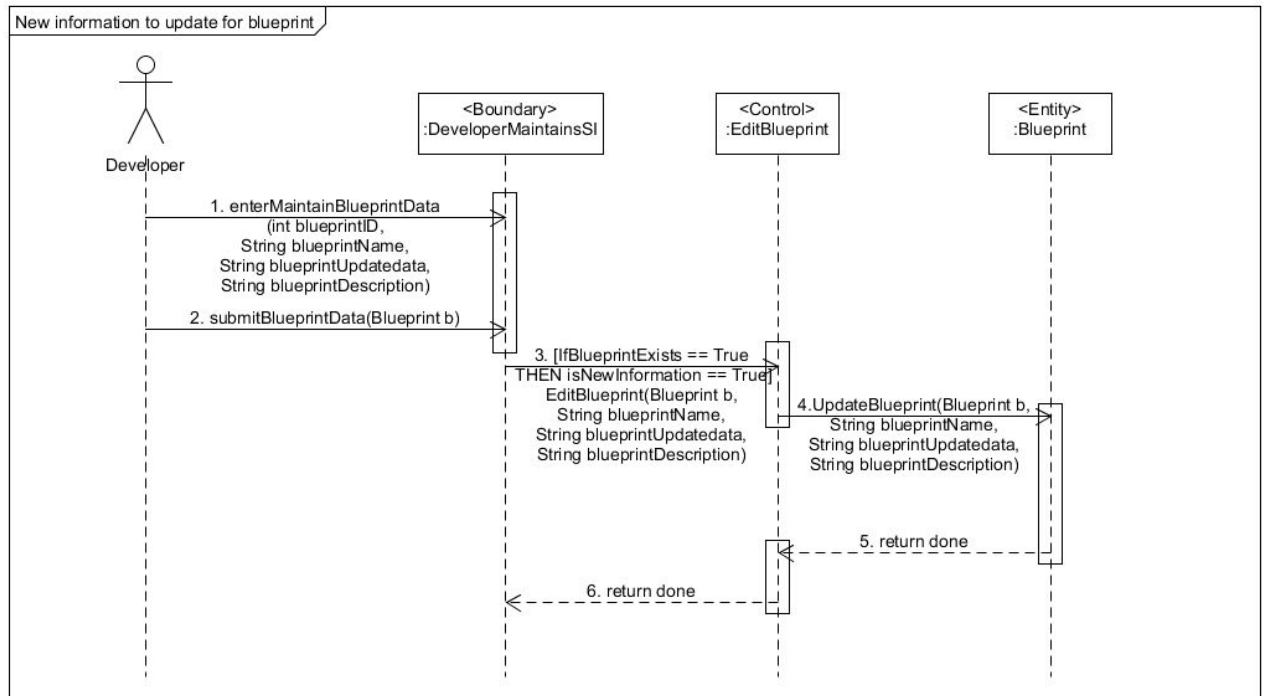
Scenario 4: Blueprint to add already exists



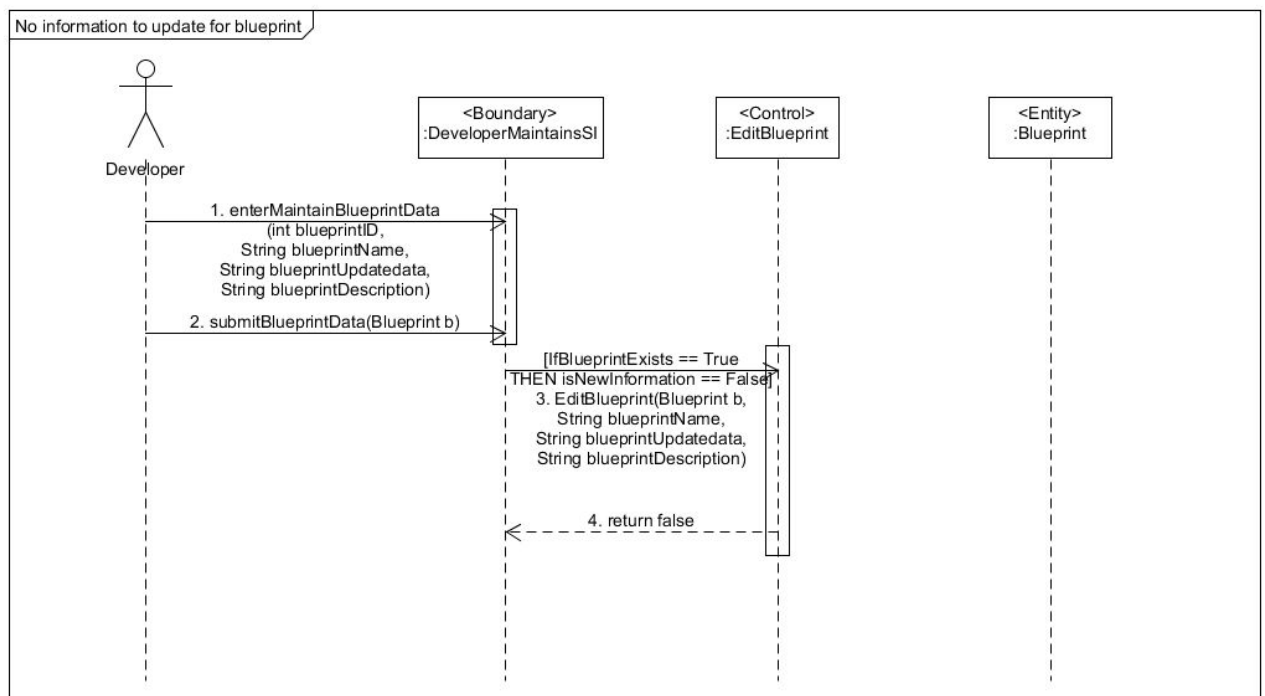
Use-Case Name: 3.3 Edit Blueprint

Description: In this use case, it talks about the role of the Developer to the development and maintenance of the MaroonPrint application, specifically the part of the role to edit the digital version of the blueprint in the database.

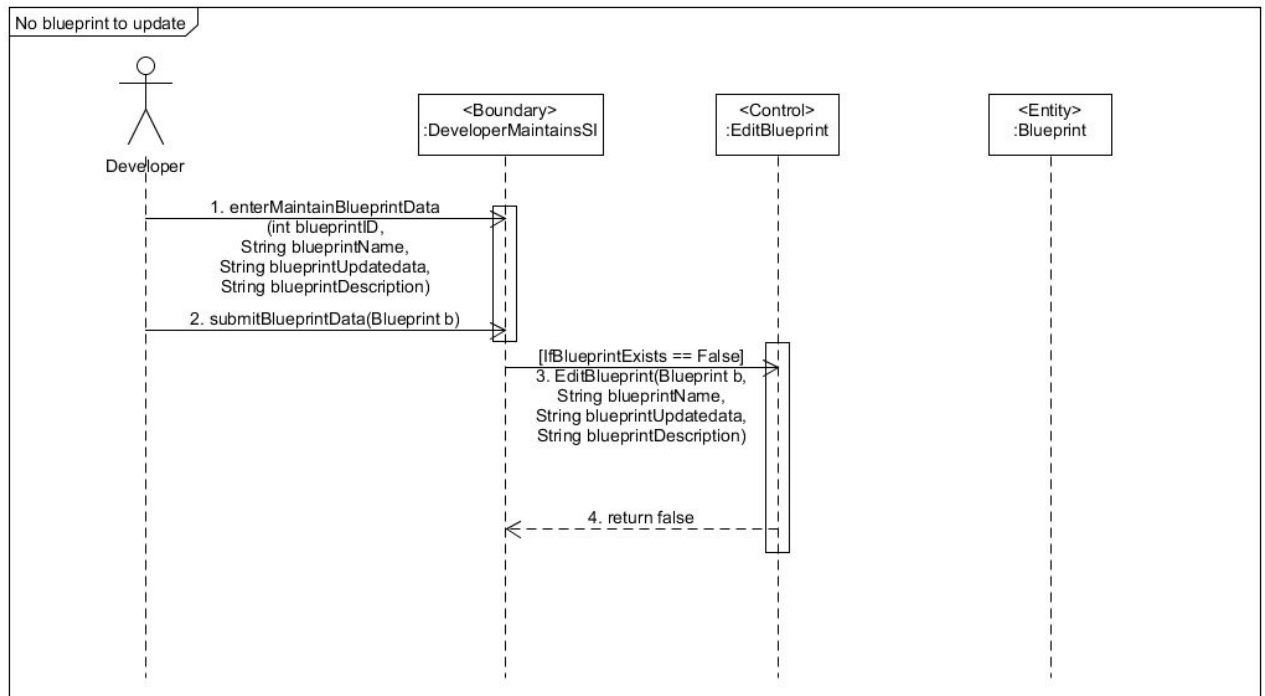
Scenario 1: Basic Flow



Scenario 2: No new information



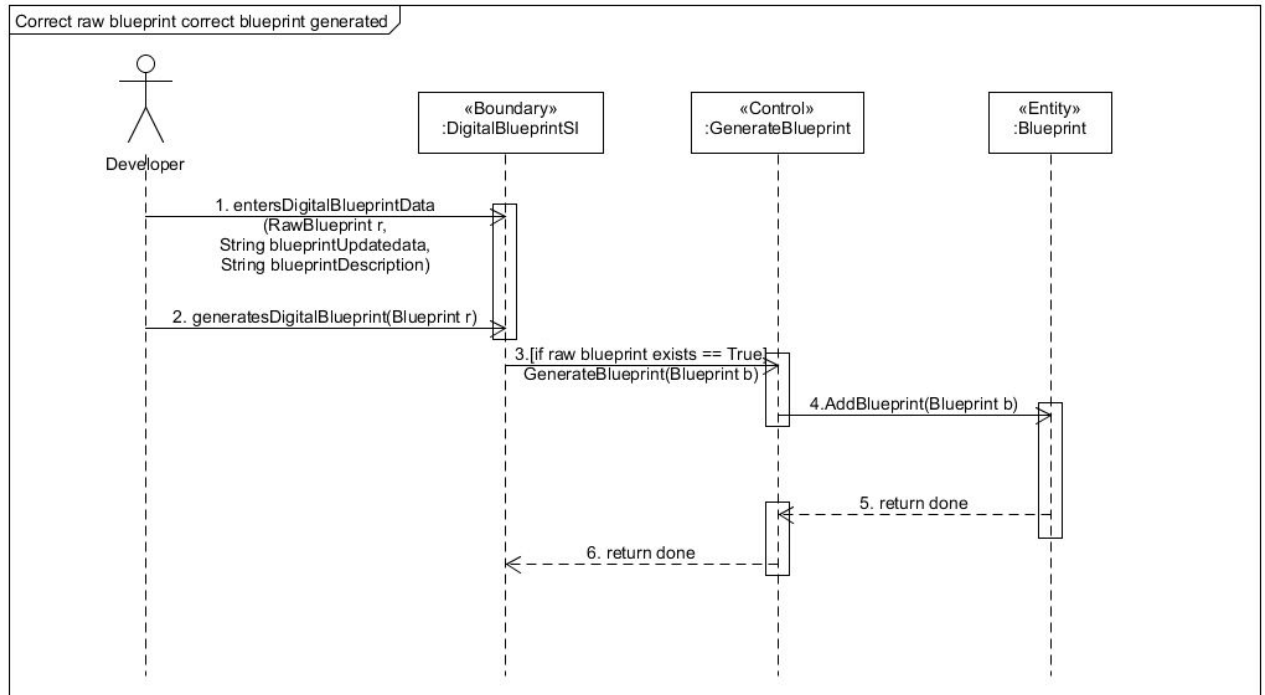
Scenario 3: No blueprint to update



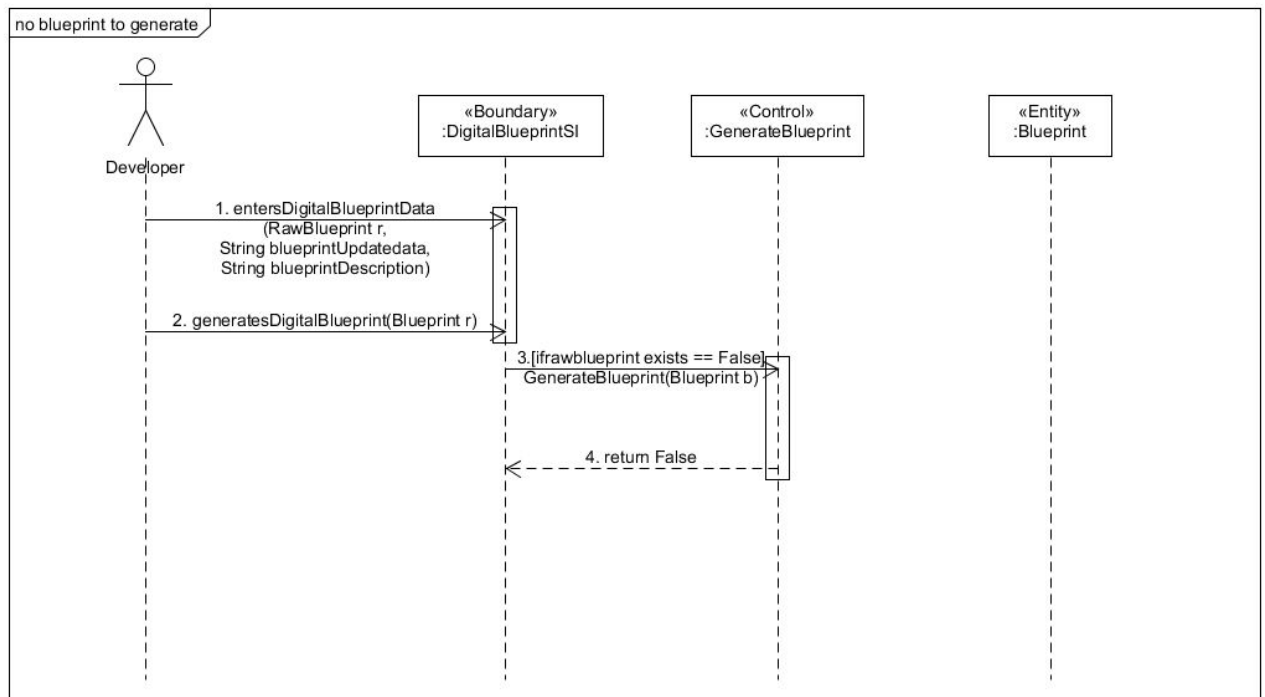
Use-Case Name: 4.0-Generates Digital Blueprint

Description: In this use case, it talks about the role of the Developer to the development and maintenance of the MaroonPrint application, specifically the part of the role to generate the digital version of the raw blueprint to add to the database.

Scenario 1: Basic Flow



Scenario 2: No blueprint to generate



3. Error in Generating Blueprint/ Making an incorrect blueprint

