**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Subsystem Design**

**for**

**The Automatic**

**Attendance Checking System**

**Version 1.1**

**Prepared by Huynh Vinh Nam**

**Le Huy Duc**

**Cao Phuong Linh**

**OOAD Group 2**

**21-Dec-2018**

**Table of Contents**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Huynh Vinh Nam | 21-Dec-2018 | Create document template | 1.0 |
| Huynh Vinh Nam | 22-Dec-2018 | Add Picture from model file | 1.1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

**1.1. Purpose**

This is a report on the subject Object-oriented Analysis and Design of group two, class ICT-BI7 about Run-time Architecture Solution.

The report is written based on the reporting format “IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications”. Content-based report is accepted and be satisfied with group meeting minute on 22-Dec-2018 (Facebook messenger).

**1.2. Intended Audience and Reading Suggestions**

*The different types of reader that the document is intended for are:*

**● Project managers:** who manage and take respond for the quality of the system. Project

managers should read the whole document for planning and assigning work.

**● Developers:** Dev is the person who implement the system from the design and documents into a runnable version. Dev have to read the whole document to implement the right system.

**● Documentation writers:** who will write the future document (report, minutes).

Documentation writers should read to understand the Use Case Main Diagram part.

This section describes the Subsystem Design Solution. Design Subsystems are used to encapsulate behavior inside a “package” which is provides explicit and formal interfaces, and which (by convention) does not expose any of its internal contents. It is used as a unit of behavior in the system, which provides the ability to completely encapsulate the interactions of a number of class and/or subsystems.

**1.3. Product Scope**

The software’s main users are students and lecturers. Software will create an environment where user (student) can check for the attendance and user (lecturer) can view and/or manage the attendance list in the course(s).

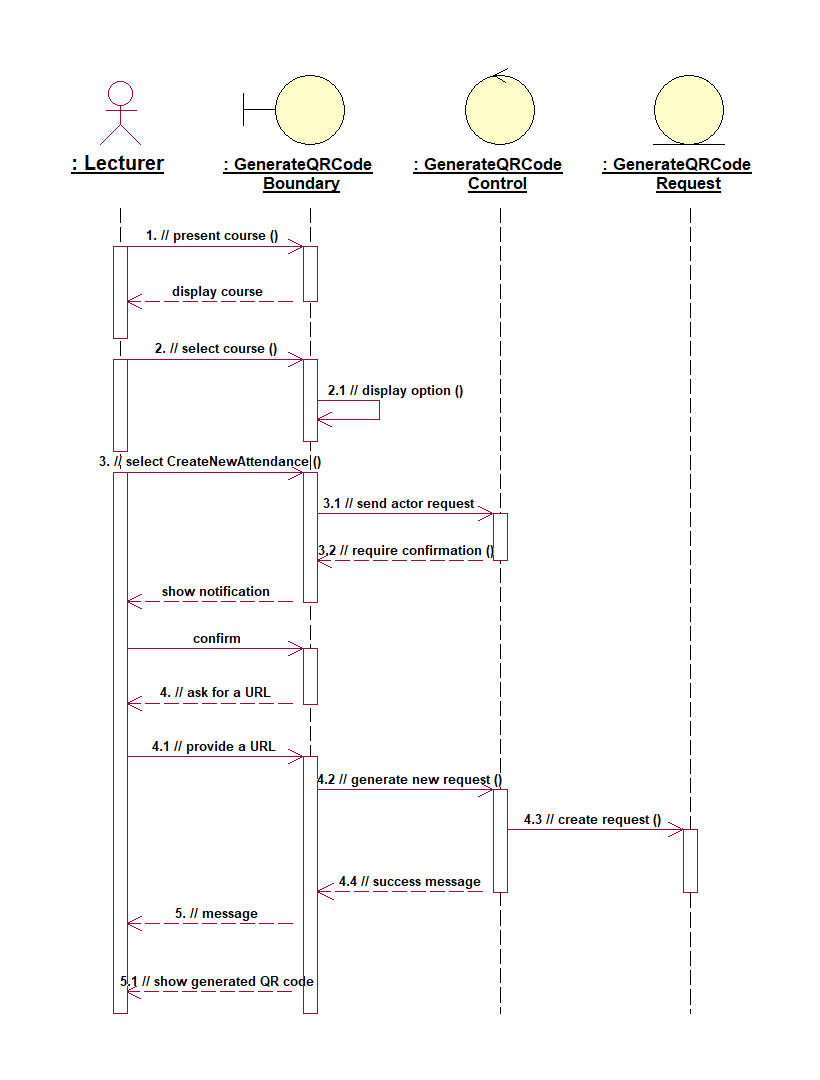
**1.4. References**

[1] Form of presentation IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

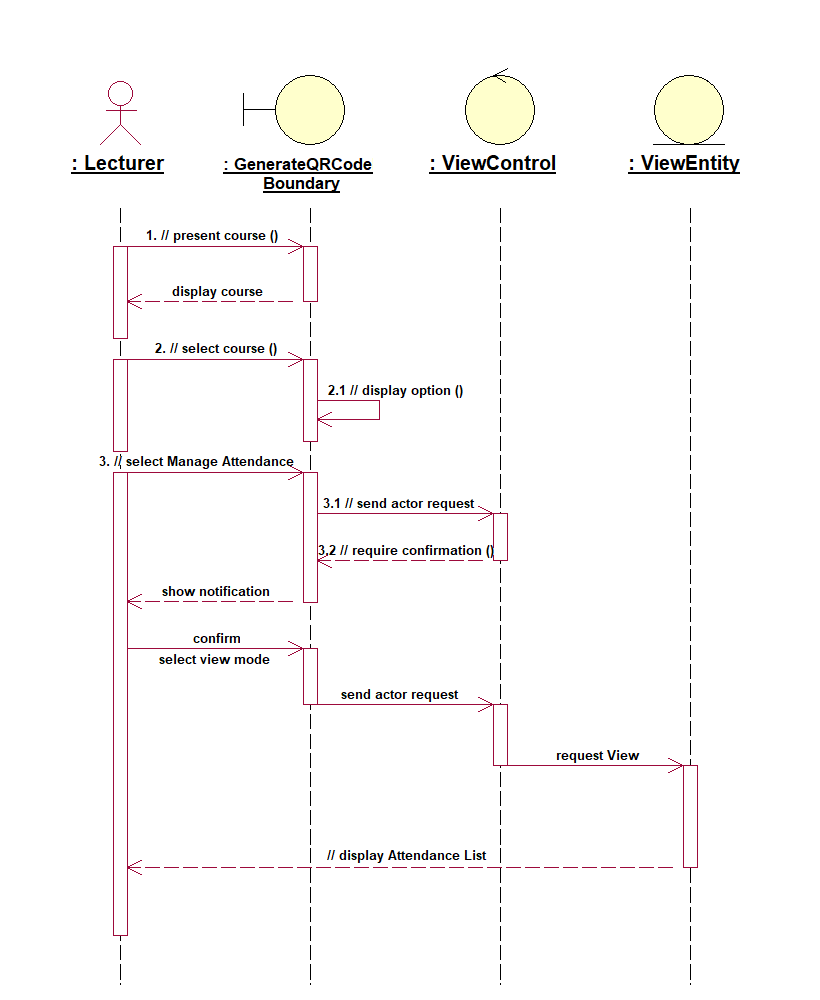
**2. Subsystem Design**

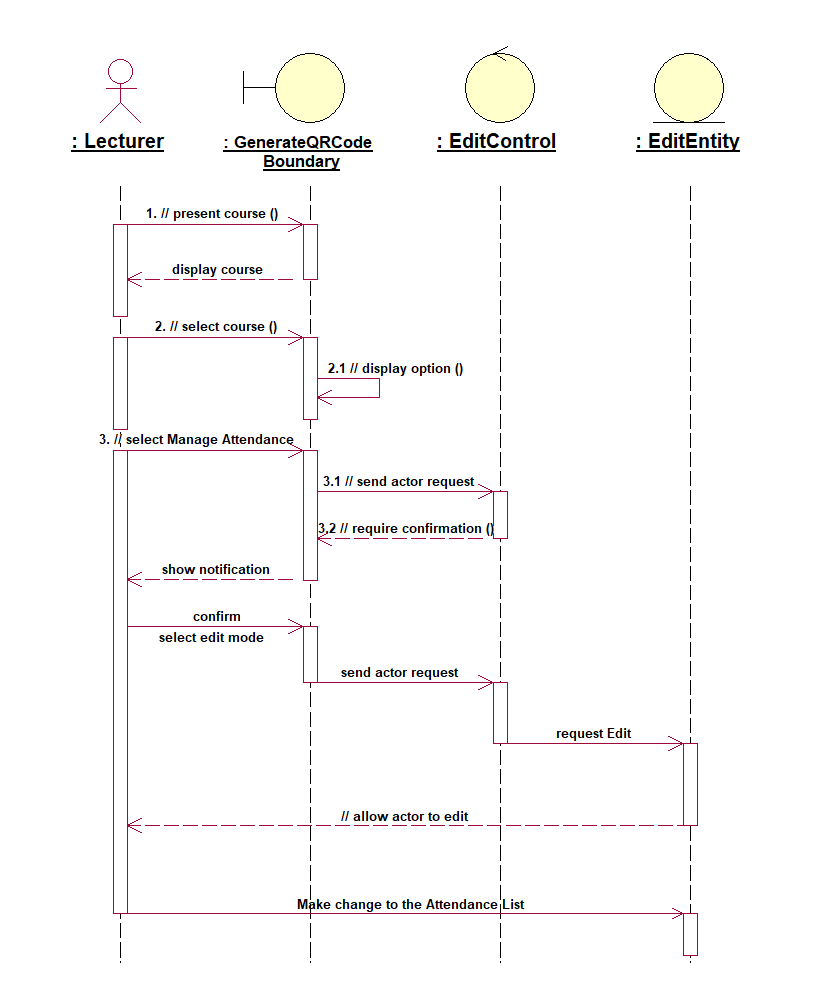
**2.1. Lecture System**

**2.1.1 Interface Realizations - Lecture System - Generate QR Code**

****

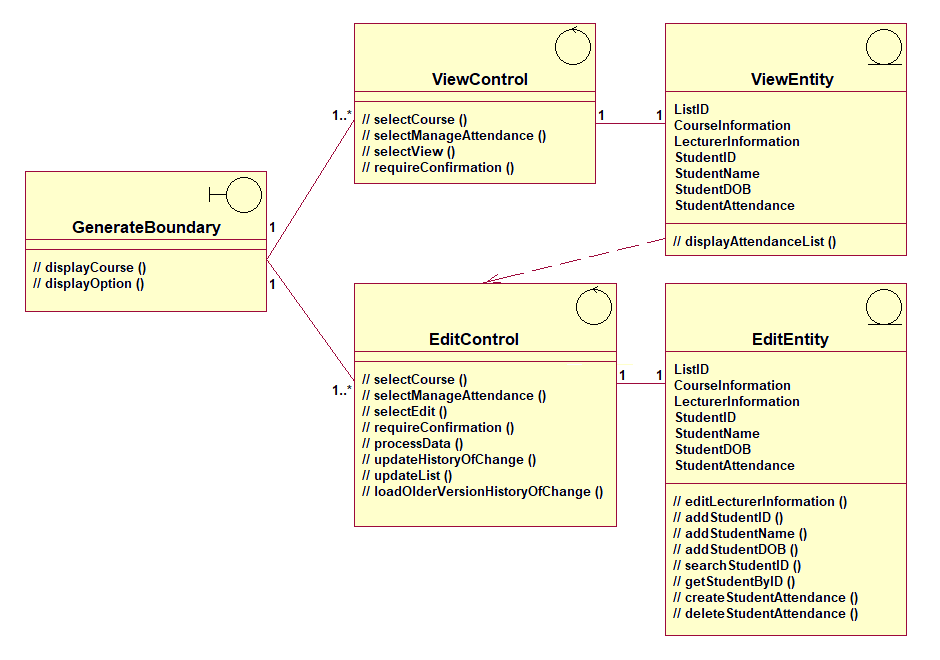
*Figure 1: Lecture System - Generate QR Code Sequence Diagram*

**2.1.1 Interface Realizations - Lecture System - Manage Attendance checklist**

****

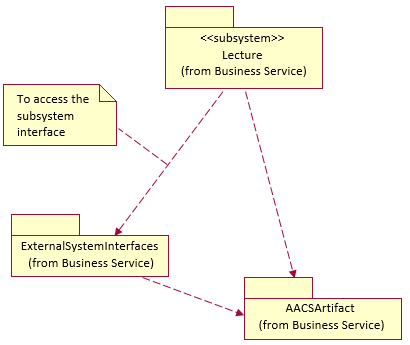
*Figure 2: Lecture System - Edit checklist Sequence Diagram*

**2.1.2 Interface Realizations - Lecture System - VOPC**

****

*Figure 3: Lecture System - VOPC Diagram*

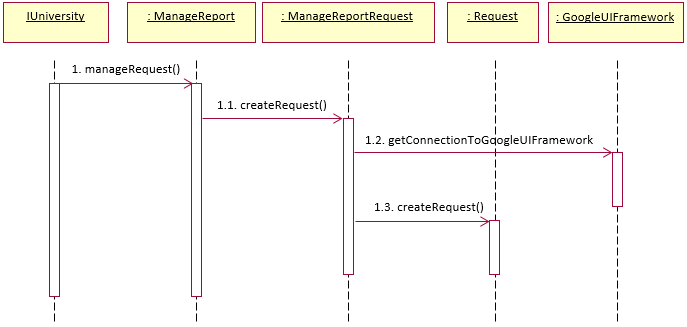
**2.1.3 Subsystem Dependencies Class Diagram**

****

*Figure 4: Lecture System Dependencies Class Diagram*

**2.2. University System**

**2.2.1 Interface Realizations - University System - Manage Report**

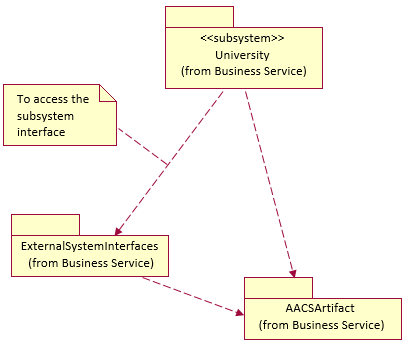
**

*Figure 5: University System - Manage Report Sequence Diagram*

**2.2.2 Interface Realizations - University System - Manage Report - VOPC**

*Figure 6: University System - VOPC Diagram*

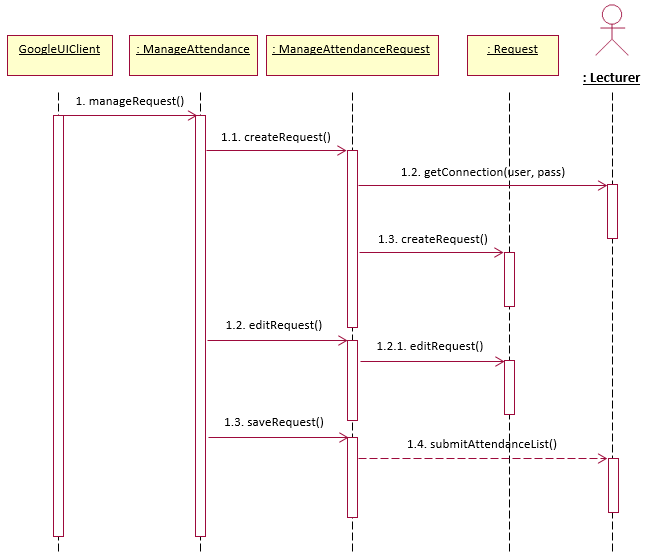
**2.2.3 Subsystems Dependency Class Diagram**



*Figure 7: University Subsystem Dependencies Class Diagram*

**2.3. Manage Attendance Checklist System**

**2.3.1 Interface Realizations - Manage Attendance checklist System**

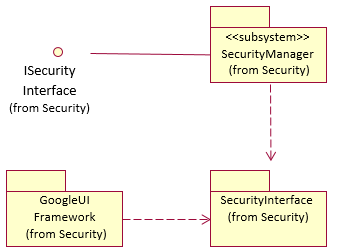
****

*Figure 8: Manage Attendance checklist System Sequence Diagram*

**2.3.2 Interface Realizations - Manage Attendance checklist System - VOPC**

*Figure 9: Manage Attendance checklist System - VOPC Diagram*

**2.3.3 Subsystems Dependency Class Diagram**

**

*Figure 10: Manage Attendance checklist Subsystem Dependencies Class Diagram*