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

**Use Case Model**

**for**

**The Automatic**

**Attendance Checking System**

**Version 1.0**

**Prepared by Huynh Vinh Nam**

**Le Huy Duc**

**Cao Phuong Linh**

**OOAD Group 2**

**25-Nov-2018**

**Table of Contents**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Huynh Vinh Nam | 25-Nov-2018 | Create document template | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

**1.1. Purpose**

This is a report on the subject Object-oriented Analysis and Design of group two, class ICT-BI7 about Use Case Model of the project AACS.

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 27-Nov-2018 at room 318, 2A of VAST, Hanoi.

**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.

**● Tester:** Tester should read the detail to write unit test particularly.

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

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

*The content of report includes two main parts:*

**● Part 1 Use Case Diagram:** The first part is a Use Case main diagram of “The Automatic Attendance Checking System”. The diagram includes: the Actors, the Use Case and the relationships between the Actors and the Use Cases.

**● Part 2 Detail of the Use Case:** Each Use Case has one corresponding description. In the description of each Use Case, we have brief description, flow of events (basic flow and alternative flows), special requirements, pre-condition, post-condition and extension points. These are enough to specify the Use Case.

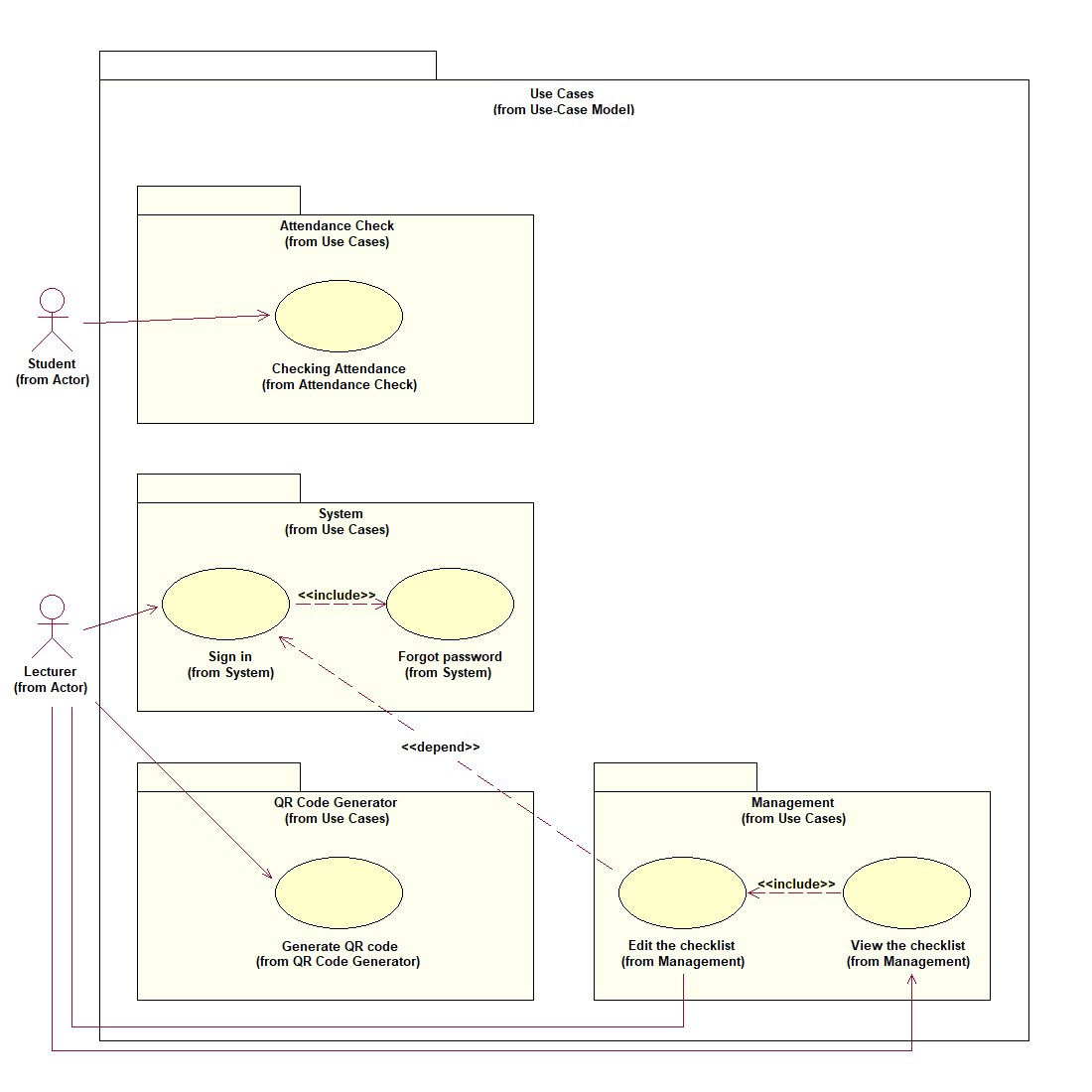
**1.3. Product Scope**

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**

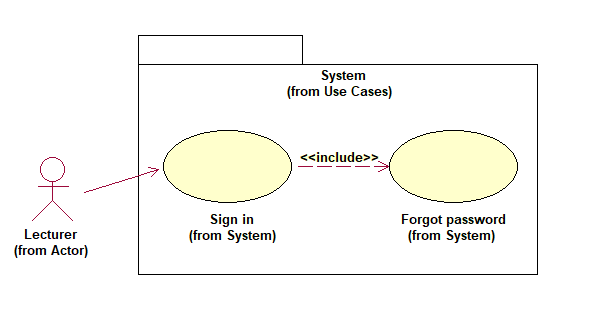
[1] Form of presentation IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

**2. Use Case Main Diagram**

****

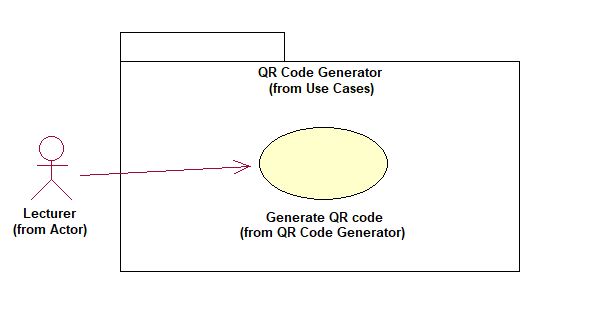
*Figure 1: Use Case Main Diagram*

**2.1. Use Case System Package Diagram**

****

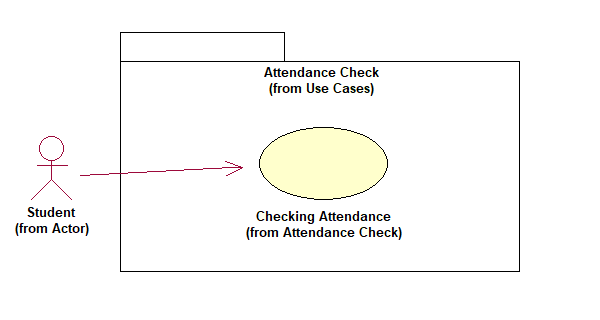
*Figure 2: Use Case System Package Diagram*

**2.2. Use Case QR Code Generator Package Diagram**

****

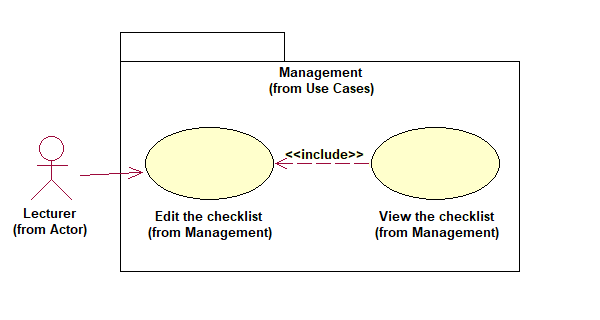
*Figure 3: Use Case QR Code Generator Package Diagram*

**2.3. Use Case Attendance Check Package Diagram**

****

*Figure 4: Use Case Attendance Check Package Diagram*

**2.4. Use Case Management Package Diagram**

****

*Figure 5: Use Case Management Package Diagram*

**3. Use Case Model Description**

**3.1. Actors**

The environment of the system includes 2 Actors: Student, Lecturer

**3.2. Use Cases**

**3.2.1. System Package**

System package includes all Use Cases related to interaction between Actor and system. System package has 2 Use Cases:

**●** Sign in

**●** Forgot password (included in Sign in)

**3.2.2. QR Code Generator Package**

QR Code Generator package includes all Use Cases related to the generation of QR code. QR Code Generator package has 1 Use Cases:

**●** Generate QR code

**3.2.3. Attendance Check Package**

Attendance Check package includes all Use Cases related to the checking attendance activity. Attendance Check package has 1 Use Cases:

**●** Checking attendance

**3.2.4. Management Package**

Management package includes all Use Cases related to manage the attendance list(s). Management package has 2 Use Cases:

**●** View the checklist

**●** Edit the checklist (included in View the checklist)

**4. Use Case Specification**

**4.1. Sign In**

**4.1.1. Brief Description**

The Use Case describes how a user logs into the Automatic Attendance Checking System.

**4.1.2. Flow of Events**

***4.1.2.1. Basic Flow***

This Use Case starts when the Actor wishes to Login to the Automatic Attendance Checking System.

|  |  |  |
| --- | --- | --- |
| **Actor Action** | **System Reaction** | **Data** |
|  | 1. The system requests that the Actor enter his/her email and password. | **●** Email\*  **●** Password\*  \* Required fields |
| 2. The Actor enters his/her email and password [or select **forgot password** to restore his/her password in other case. The **forgot password** sub flow is executed.] | 3. The system validates the entered name and password and logs the Actor into the system. |  |

*4.1.2.1.1 Forgot Password*

At Step 3:

|  |  |  |
| --- | --- | --- |
| **Actor Action** | **System Reaction** | **Data** |
| 1. The Actor select forgot password. | 2. The system shows alert and requests that the Actor enter his/her email. | **●** Email\*  \* Required fields |
| 3. The Actor enters his/her email. | 4. The system validates the entered email and shows notification to the Actor. |  |

***4.1.2.2. Alternative Flow***

*4.1.2.2.1 Invalid Email/Password*

If, in the **Basic Flow**, at step 2, the Actor enters an invalid email and/or password, the system

displays an error message. The Actor can choose to either return to the beginning of the **Basic Flow** or cancel the login, at which point the Use Case ends.

*4.1.2.2.2 Email doesn’t exist*

If, in the **Forgot Password** sub flow, at step 3, the Actor enters an email that doesn’t exist on the system, the system displays an error message. The Actor can go back to either the beginning of the **Basic Flow** or beginning of the **Forgot Password** sub flow.

**4.1.3 Special Requirement**

None.

**4.1.4 Pre-Condition**

None.

**4.1.5 Post-Condition**

If the Use Case was successful, the Actor is now logged into the system.

Otherwise, the system state is unchanged.

**4.1.6 Extension Points**

None.

**4.2. Generate QR Code**

**4.2.1. Brief Description**

The Use Case allows a logged-user (a lecturer) provide a URL that redirect to the attendance form and the system should generate QR code base on the given link automatically.

**4.2.2. Flow of Events**

***4.2.2.1. Basic Flow***

This Use Case starts when a lecturer wants to generate a QR Code for students to check their attendance in his/her current lecture.

|  |  |  |
| --- | --- | --- |
| **Actor Action** | **System Reaction** | **Data** |
| 1. The Actor selects his/her course. | 2. The system receives the information then redirects the Actor to two options: “Create a new attendance for today” or “Manage the attendance checklist”. |  |
| 3. The Actor selects the first option. | 4. The system asks for a URL so that it can automatically generate QR code. | **●** URL\*  \* Required fields |
|  | 5. The system sends a successful message to the Actor and then shows the generated QR code |  |

***4.2.2.2. Alternative Flow***

*4.2.2.2.1 The second option selected*

If, in the **Basic Flow**, at step 3, the Actor chooses “Manage the attendance checklist”, the system reask the Actor to confirm if he/she sure want to go to manage session. The Actor can either go back to the step 3 of the **Basic Flow** or be redirected to the **Management Package**, at which point the Use Case ends.

**4.2.3 Special Requirement**

None.

**4.2.4 Pre-Condition**

The Actor must be logged into the system.

**4.2.5 Post-Condition**

If the Use Case was successful, a new QR Code is created for the Actor.

Otherwise, the system state is unchanged.

**4.2.6 Extension Points**

None.

**4.3. Checking Attendance**

**4.3.1. Brief Description**

The Use Case allows a student to checking attendance the system should update the checklist automatically.

**4.3.2. Flow of Events**

***4.3.2.1. Basic Flow***

This Use Case starts when a student wishes to check for his/her attendance in the current lecture.

|  |  |  |
| --- | --- | --- |
| **Actor Action** | **System Reaction** | **Data** |
| 1. The Actor wishes to check attendance. | 2. The system displays the QR code that the lecturer provided. |  |
| 3. The Actor accesses the URL (by scan QR code) then confirms for his/her name in the form provided in the link. | 4. The system retrieves student’s information. | **●** Student’s ID\*  \* Required fields |
|  | 5. The system updates the checklist by giving a mark on attendance for that Actor. | **●** Student’s ID  **●** Student’s Name  **●** Course |

***4.3.2.2. Alternative Flow***

*4.3.2.2.1 The Actor neither can scan the generated QR code nor access the URL*

If, in the **Basic Flow**, at step 3, the Actor can’t provide his/her own informations to the system, he/she should inform the lecturer to alter the checklist at the end of the current lecture.

**4.3.3 Special Requirement**

None.

**4.3.4 Pre-Condition**

The QR Code is provided by the lecturer.

**4.3.5 Post-Condition**

If the Use Case was successful, the Actor is approved as already attended into the system.

Otherwise, the system state is unchanged.

**4.3.6 Extension Points**

None.

**4.4. View the checklist**

**4.4.1. Brief Description**

The Use Case allows a lecturer to view the attendance list.

**4.4.2. Flow of Events**

***4.4.2.1. Basic Flow***

This Use Case starts when a lecturer wants to view the attendance checklist.

|  |  |  |
| --- | --- | --- |
| **Actor Action** | **System Reaction** | **Data** |
| 1. The Actor selects his/her course. | 2. The system receives the information then redirects the Actor to two options: “Create a new attendance for today” or “Manage the attendance checklist”. |  |
| 3. The Actor selects the second option. | 4. The system asks the Actor if he/she want to view or making change to the list |  |
| 5. The Actor selects view mode [or select **edit mode** to alter the list. The **edit mode** sub flow is executed.] | 6. The system displays the checklist to the Actor |  |

*4.4.2.1.1 Edit Mode*

At Step 5:

|  |  |  |
| --- | --- | --- |
| **Actor Action** | **System Reaction** | **Data** |
| 1. The Actor chooses to make change to the list. | 2. The system shows alert and requests the Actor to confirm for this action. |  |
| 3. The Actor verifies the last action. | 4. The system redirect the Actor to the editable list. |  |
| 5. The Actor performs some changes | 5. The system adds these new informations to the history of changes. | **●** Student’s ID  **●** Attendance mark |

***4.4.2.2. Alternative Flow***

*4.4.2.2.1 The first option selected*

If, in the **Basic Flow**, at step 3, the Actor chooses “Create a new attendance for today”, the system reask the Actor to confirm if he/she sure want to generate a new QR code. The Actor can either go back to the step 3 of the **Basic Flow** or be redirected to the **QR Code Generator Package**, at which point the Use Case ends.

**4.4.3 Special Requirement**

None.

**4.4.4 Pre-Condition**

The Actor must be logged into the system.

**4.4.5 Post-Condition**

*Basic Flow (View option):* If the Use Case was successful, the Actor is able to view the checklist provided by the system.

*Sub Flow (Edit option):* If the Use Case was successful, the Actor is able to edit the checklist provided by the system.

Otherwise, the system state is unchanged.

**4.4.6 Extension Points**

None.