**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

Equipment’s Classroom Management

|  |  |
| --- | --- |
| Group 10 | |
| **Group member** | Tran Vinh Quang – SE61078  Tang Viet Hung – SE61019  Doan Nguyen Minh Chi – SE60717 |
| **Supervisor** | Kieu Trong Khanh |
| **Ext. Supervisor** | N/A |
| **Capstone Project Code** | ECRM |

-Ho Chi Minh City, May 11, 2015-

Table of Contents

[Table of Contents 3](#_Toc426362750)

[**List of Tables** 11](#_Toc426362751)

[List of Figures 12](#_Toc426362752)

[Definitions, Acronyms and Abbreviations 13](#_Toc426362753)

[**A.** **Introduction** 14](#_Toc426362754)

[**1.** **Project Information** 14](#_Toc426362755)

[**2.** **Introduction** 14](#_Toc426362756)

[**3.** **Current Situation** 14](#_Toc426362757)

[**4.** **Problem Definition** 14](#_Toc426362758)

[**5.** **Proposed Solution** 14](#_Toc426362759)

[**5.1.** **Feature Function** 15](#_Toc426362760)

[**5.2.** **Advantages and disadvantages** 16](#_Toc426362761)

[**6.** **Functional Requirement** 16](#_Toc426362762)

[**6.1.** **Equipment Management** 16](#_Toc426362763)

[**6.2.** **Notification** 16](#_Toc426362764)

[**6.3.** **Room Type Management** 16](#_Toc426362765)

[**6.4.** **Classroom Management** 16](#_Toc426362766)

[**6.5.** **Damaged Analysis And Recommendations** 17](#_Toc426362767)

[**6.6.** **Suggest Available Classroom** 17](#_Toc426362768)

[**6.7.** **Manage Account** 17](#_Toc426362769)

[**6.8.** **Manage Equipment Category** 17](#_Toc426362770)

[**6.9.** **Tracking Schedule** 17](#_Toc426362771)

[**7.** **Role and Responsibility** 17](#_Toc426362772)

[**B.** **Software Project Management Plan** 18](#_Toc426362773)

[**1.** **Problem Definition** 18](#_Toc426362774)

[**1.1.** **Name of this Capstone Project** 18](#_Toc426362775)

[**1.2.** **Problem Abstract** 18](#_Toc426362776)

[**1.3.** **Project Overview** 18](#_Toc426362777)

[**1.3.1.** **Current Situation** 18](#_Toc426362778)

[**1.3.2.** **The proposed system** 18](#_Toc426362779)

[**1.3.2.1.** **Website application** 18](#_Toc426362780)

[**1.3.2.2.** **Mobile application for teacher (ECRM Teacher)** 19](#_Toc426362781)

[**1.3.2.3.** **Mobile application for staff (ECRM Staff)** 19](#_Toc426362782)

[**1.3.3.** **Boundaries of the System** 20](#_Toc426362783)

[**1.3.4.** **Development Environment** 20](#_Toc426362784)

[**1.3.4.1.** **Hardware requirements** 20](#_Toc426362785)

[**1.3.4.2.** **Software requirements** 20](#_Toc426362786)

[**2.** **Project organization** 20](#_Toc426362787)

[**2.1.** **Software Process Model** 20](#_Toc426362788)

[**2.2.** **Roles and responsibilities** 21](#_Toc426362789)

[**2.3.** **Tools and Techniques** 22](#_Toc426362790)

[**3.** **Project Management Plan** 22](#_Toc426362791)

[**3.1.** **Software development life cycle** 22](#_Toc426362792)

[**3.2.** **Phase Detail** 24](#_Toc426362793)

[**3.2.1.** **Phase 1: Room Type Management.** 24](#_Toc426362794)

[**3.2.2.** **Phase 2: Classroom Management.** 25](#_Toc426362795)

[**3.2.3.** **Phase 3: Equipment Management.** 25](#_Toc426362796)

[**3.2.4.** **Phase 4: Account Management.** 26](#_Toc426362797)

[**3.2.5.** **Phase 5: Tracking Schedule.** 27](#_Toc426362798)

[**3.2.6.** **Phase 6: Report Management** 28](#_Toc426362799)

[**3.2.7.** **Phase 7: Web Service** 28](#_Toc426362800)

[**3.2.8.** **Phase 8: Android Application** 29](#_Toc426362801)

[**3.3.** **All Meeting Minutes** 30](#_Toc426362802)

[**4.** **Coding Convention** 30](#_Toc426362803)

[**C.** **Software Requirement Specification** 31](#_Toc426362804)

[**1.** **User Requirement Specification** 31](#_Toc426362805)

[**1.1.** **Guest Requirement** 31](#_Toc426362806)

[**1.2.** **User Requirement** 31](#_Toc426362807)

[**1.3.** **Staff Requirement** 31](#_Toc426362808)

[**1.4.** **Administrator Requirement** 32](#_Toc426362809)

[**2.** **System Requirement Specification** 32](#_Toc426362810)

[**2.1.** **External Interface Requirement** 32](#_Toc426362811)

[**2.1.1.** **User Interface** 32](#_Toc426362812)

[**2.1.2.** **Hardware Interface** 32](#_Toc426362813)

[**2.1.3.** **Software Interface** 32](#_Toc426362814)

[**2.1.4.** **Communication Protocol** 32](#_Toc426362815)

[**2.2.** **System Overview Use Case** 32](#_Toc426362816)

[**2.3.** **List of Use Case** 33](#_Toc426362817)

[**2.3.1.** **<Guest> Overview Use Case** 33](#_Toc426362818)

[**2.3.2.** **<Guest> Login** 34](#_Toc426362819)

[2.3.3. <User> Overview Use Case 36](#_Toc426362820)

[2.3.4. <User> View Report History 36](#_Toc426362821)

[2.3.5. <User> View Room Map 38](#_Toc426362822)

[2.3.6. <User> Create Report 39](#_Toc426362823)

[2.3.7. <Staff> Use Case Overview 42](#_Toc426362824)

[2.3.8. <Staff> Create New Room Type 43](#_Toc426362825)

[2.3.9. <Staff> Update Room Type 46](#_Toc426362826)

[2.3.10. <Staff> Remove Room Type 50](#_Toc426362827)

[2.3.11. <Staff> Create Classroom 52](#_Toc426362828)

[2.3.12. <Staff> Update Classroom 54](#_Toc426362829)

[2.3.13. <Staff> Remove Classroom 55](#_Toc426362830)

[2.3.14. <Staff> Resolve Report 57](#_Toc426362831)

[2.3.16. <Staff> View Statistic 62](#_Toc426362832)

[2.3.17. <Staff> Mapping Schedule Manual 63](#_Toc426362833)

[2.3.18. <Staff> Mapping Schedule Import File 65](#_Toc426362834)

[**2.3.19.** **<Authorized User> Use case overview** 66](#_Toc426362835)

[2.3.20. <Authorized User> Logout 67](#_Toc426362836)

[**2.3.21.** **<Authorized User> View Profile** 68](#_Toc426362837)

[2.3.22. <Authorized User> Update Account 70](#_Toc426362838)

[**2.3.23.** **<Administrator> Use case Overview** 72](#_Toc426362839)

[**2.3.24.** **< Administrator> Add Account** 72](#_Toc426362840)

[2.3.25. <Administrator> Activate/Deactivate Account 74](#_Toc426362841)

[2.3.26. <Administrator> Update Account 76](#_Toc426362842)

[**2.3.27.** **<Administrator> Remove Account** 78](#_Toc426362843)

[**2.3.28.** **<System> Use case Overview** 79](#_Toc426362844)

[**2.3.29.** **<System> Notify about fixed** 80](#_Toc426362845)

[**2.3.30.** **<System> Send SMS** 82](#_Toc426362846)

[**2.3.31.** **<System> Suggest Available Room** 83](#_Toc426362847)

[**2.3.32.** **<System> Notify about damaged** 84](#_Toc426362848)

[3. Software System Attribute 85](#_Toc426362849)

[**3.1.** **Usability** 85](#_Toc426362850)

[**3.1.1.** **Graphic User Interface** 85](#_Toc426362851)

[**3.1.2.** **Usability** 85](#_Toc426362852)

[**3.1.3.** **Installation** 86](#_Toc426362853)

[**3.2.** **Reliability** 86](#_Toc426362854)

[**3.3.** **Availability** 86](#_Toc426362855)

[**3.4.** **Security** 86](#_Toc426362856)

[**3.5.** **Maintainability** 86](#_Toc426362857)

[**3.6.** **Portability** 86](#_Toc426362858)

[**3.7.** **Performance** 86](#_Toc426362859)

[**4.** **Conceptual Diagram** 86](#_Toc426362860)

[**D.** **Software Design Description** 88](#_Toc426362861)

[**1.** **Design Overview** 88](#_Toc426362862)

[**2.** **System Architecture Design** 88](#_Toc426362863)

[**2.1.** **Web Application architecture description** 89](#_Toc426362864)

[**2.2.** **Mobile Application architecture description** 90](#_Toc426362865)

[**3.** **Component Diagram** 90](#_Toc426362866)

[**4.** **Detail Description** 91](#_Toc426362867)

[**4.1.** **Class Diagram** 91](#_Toc426362868)

[**4.2.** **Class Diagram Explanation** 91](#_Toc426362869)

[**4.3.** **Interactive Diagram** 91](#_Toc426362870)

[**4.3.1.** **Web Application** 91](#_Toc426362871)

[**4.3.1.1.** **Staff** 91](#_Toc426362872)

[**4.3.1.2.** **Teacher** 92](#_Toc426362873)

[**4.3.1.3.** **Guest** 92](#_Toc426362874)

[**4.3.1.4.** **Admin** 92](#_Toc426362875)

[**4.3.2.** **Mobile Application** 92](#_Toc426362876)

[**4.3.2.1.** **ECRM Staff** 92](#_Toc426362877)

[**4.3.2.2.** **ECRM Teacher** 92](#_Toc426362878)

[**5.** **Interface** 92](#_Toc426362879)

[**5.1.** **Component Interface** 92](#_Toc426362880)

[**5.1.1.** **Web Service Interface** 92](#_Toc426362881)

[**5.2.** **Web Application Design** 96](#_Toc426362882)

[**5.2.1.** **User Interface Design** 96](#_Toc426362883)

[**5.2.2.** **Staff Interface Design** 96](#_Toc426362884)

[**5.2.3.** **Admin Interface Design** 96](#_Toc426362885)

[**5.3.** **ECRM Staff Design** 97](#_Toc426362886)

[**5.3.1.** **Login Screen** 97](#_Toc426362887)

[**5.3.2.** **List Report Newest Screen** 97](#_Toc426362888)

[**5.3.3.** **List Report On Going Screen** 97](#_Toc426362889)

[**5.3.4.** **List Report Finished Screen** 97](#_Toc426362890)

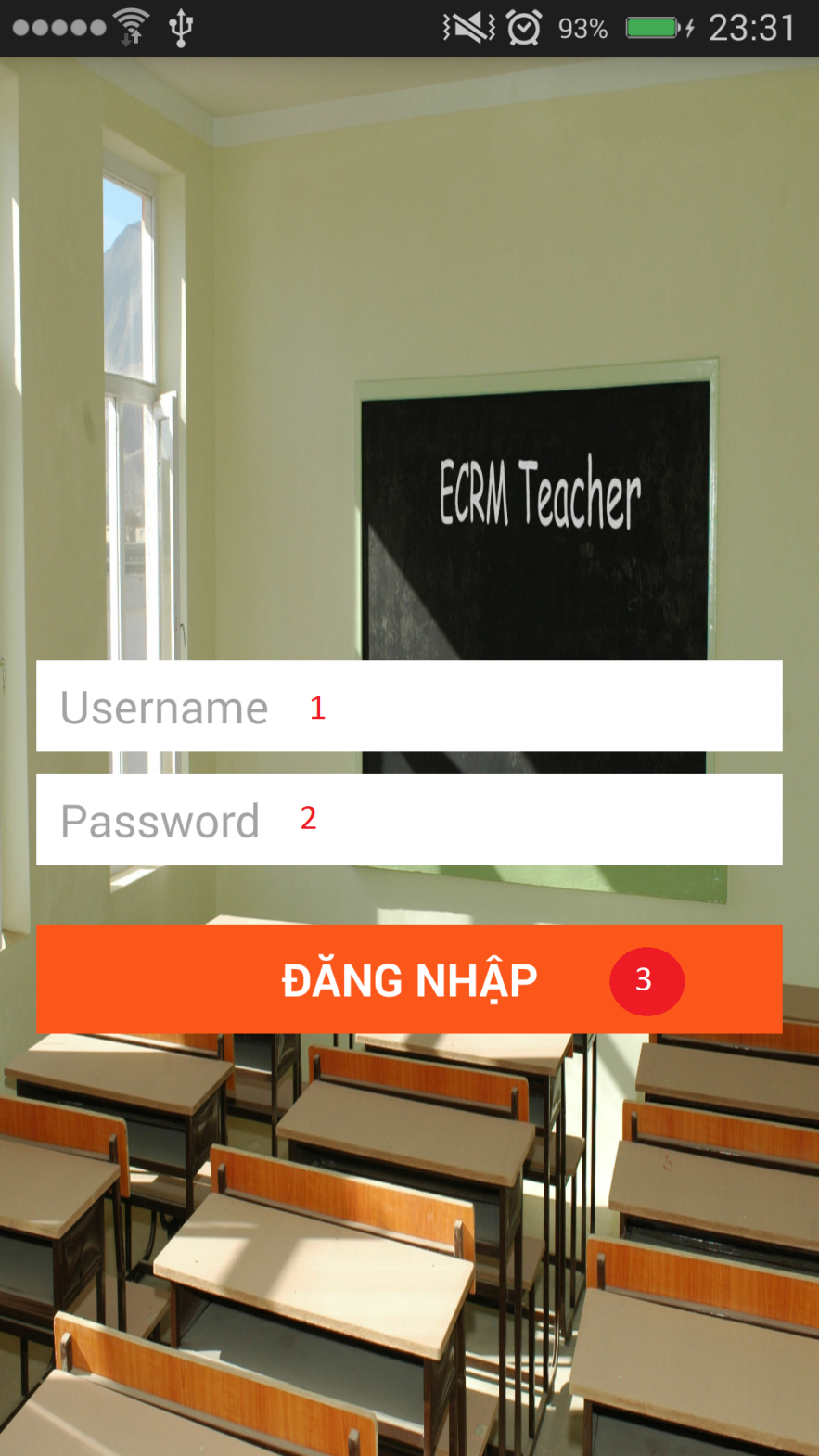
[**5.3.5.** **Report Detail Screen** 97](#_Toc426362891)

[**5.3.6.** **Change Room Screen** 97](#_Toc426362892)

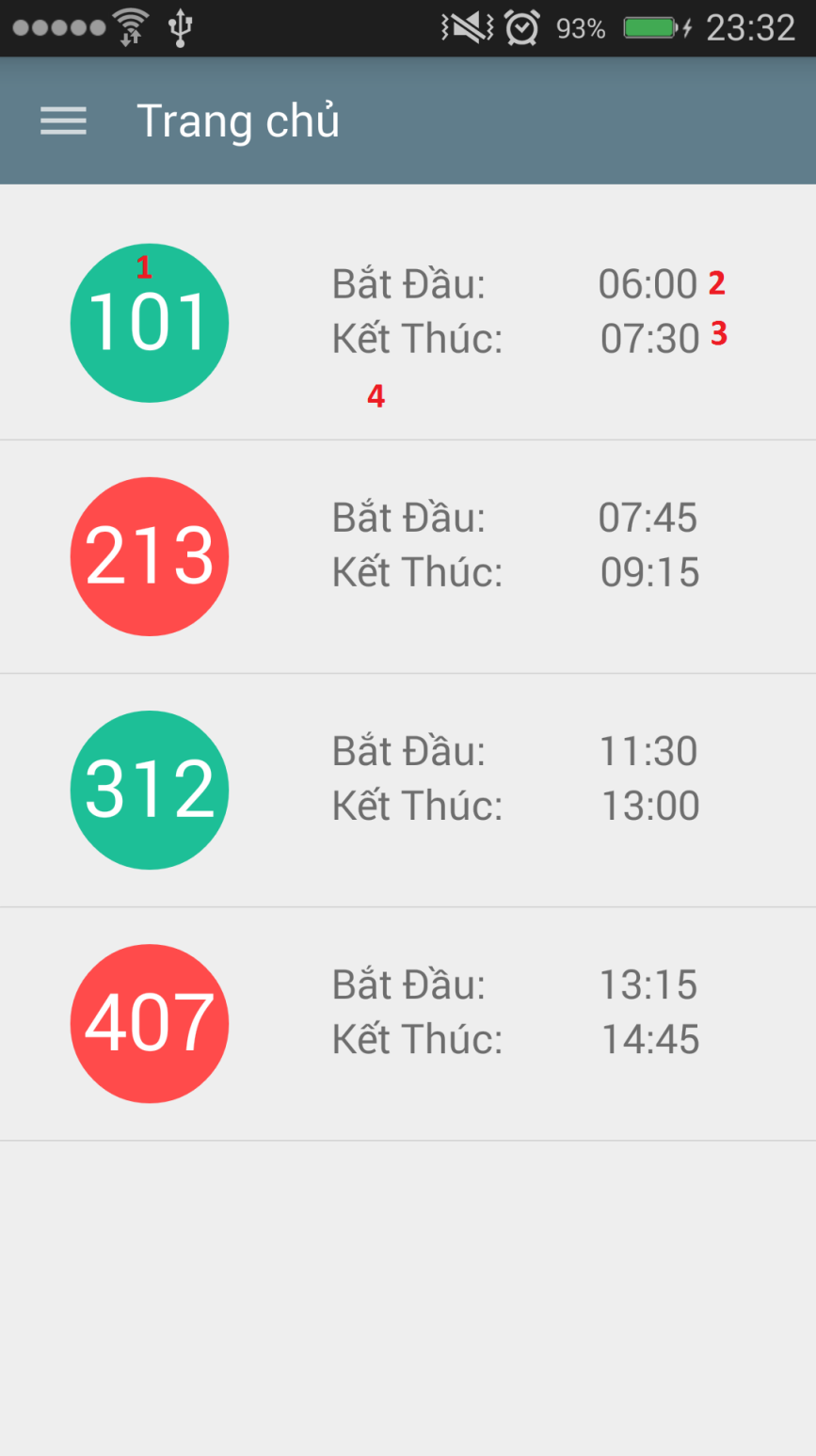
[**5.3.7.** **Resolve Report Screen** 97](#_Toc426362893)

[**5.4.** **ECRM Teacher Design** 97](#_Toc426362894)

[**5.4.1.** **Login Screen.** 97](#_Toc426362895)

[**** 98](#_Toc426362896)

[**5.4.2.** **Schedule Screen** 99](#_Toc426362897)

[**** 100](#_Toc426362898)

[**5.4.3.** **View Report History Screen** 101](#_Toc426362899)

[**5.4.4.** **Create Report Screen** 101](#_Toc426362900)

[**5.4.5.** **Change Password Screen** 101](#_Toc426362901)

[**5.4.6.** **Report Detail Screen** 101](#_Toc426362902)

[**6.** **Database Design** 101](#_Toc426362903)

[**6.1.** **Entity Relationship Diagram** 101](#_Toc426362904)

[**6.2.** **Entity Dictionary** 101](#_Toc426362905)

[**7.** **Algorithms** 101](#_Toc426362906)

[**E.** **System Implementation & Test** 102](#_Toc426362907)

[**1.** **Introduction** 102](#_Toc426362908)

[**1.1.** **Overview** 102](#_Toc426362909)

[**1.2.** **Test Approach** 102](#_Toc426362910)

[**2.** **Database Relationship Diagram** 102](#_Toc426362911)

[**2.1.** **Physical Diagram** 102](#_Toc426362912)

[**2.2.** **Data Dictionary** 102](#_Toc426362913)

[**3.** **Performance Measures** 102](#_Toc426362914)

[**3.1.** **Web Application Page Load Speed** 102](#_Toc426362915)

[**3.1.1.** **Definition** 102](#_Toc426362916)

[**3.1.2.** **Test environment** 102](#_Toc426362917)

[**3.1.3.** **Test cases** 102](#_Toc426362918)

[**3.1.4.** **Test result** 102](#_Toc426362919)

[**3.2.** **Mobile Application API Load Speed** 102](#_Toc426362920)

[**3.2.1.** **Definition** 102](#_Toc426362921)

[**3.2.2.** **Test environment** 102](#_Toc426362922)

[**3.2.3.** **Test case** 102](#_Toc426362923)

[**3.2.4.** **Test result** 102](#_Toc426362924)

[**4.** **Test Plan** 102](#_Toc426362925)

[**4.1.** **Feature to be tested** 102](#_Toc426362926)

[**4.2.** **Feature no to be tested** 102](#_Toc426362927)

[**5.** **System Testing Test Case** 102](#_Toc426362928)

**List of Tables**

[Table 1: Role and Responsibility 12](#_Toc421361158)

[Table 2: Hardware Requirement for system 15](#_Toc421361159)

[Table 3: Roles and Responsibilities Details 16](#_Toc421361160)

[Table 4: Software development life cycle 18](#_Toc421361161)

[Table 5: Phase 1: Room Type Management 19](#_Toc421361162)

[Table 6: Phase 2: Classroom Management 19](#_Toc421361163)

[Table 7: Phase 3: Equipment Management 20](#_Toc421361164)

[Table 8: Phase 4: Web Service. 20](#_Toc421361165)

[Table 9: Phase 5: Android Application 21](#_Toc421361166)

[Table 10: <Guest> Login 27](#_Toc421361167)

[Table 11: <User> View Report History 29](#_Toc421361168)

[Table 12: <User> View Room Map 30](#_Toc421361169)

[Table 13: <User> Create Report 33](#_Toc421361170)

[Table 14: <Staff> Create Room Type 37](#_Toc421361171)

[Table 15: <Staff> Update Room Type 41](#_Toc421361172)

[Table 16: <Staff> Remove Room Type 43](#_Toc421361173)

[Table 17: <Staff> Create Classroom 44](#_Toc421361174)

[Table 18: <Staff> Update Classroom 46](#_Toc421361175)

[Table 19: <Staff> Remove Classroom 48](#_Toc421361176)

[Table 20: <Staff> Resolve Report 51](#_Toc421361177)

[Table 21: <Staff> Remove Report 53](#_Toc421361178)

[Table 22: <Staff> Mapping Schedule Manual 56](#_Toc421361179)

[Table 23: <Staff> Mapping Schedule Import File 57](#_Toc421361180)

[Table 24: <Authorized User> Logout 59](#_Toc421361181)

[Table 25: <Authorized User> View Profile 61](#_Toc421361182)

[Table 26: <Authorized User> Update Account 63](#_Toc421361183)

[Table 27: <Administrator> Add New Account 65](#_Toc421361184)

[Table 28: <Administrator> Activate/Deactivate Account 67](#_Toc421361185)

[Table 29: <Administrator> Update Account 69](#_Toc421361186)

[Table 30: <Administrator> Remove Account 70](#_Toc421361187)

[Table 31: <System> Notify about fixed 73](#_Toc421361188)

[Table 32: <System> Send SMS 74](#_Toc421361189)

[Table 33: <System> Suggest Available Room 75](#_Toc421361190)

[Table 34: <System> Notify about damaged 76](#_Toc421361191)

# List of Figures

[Figure 1: Iterative Development Model 15](#_Toc421361192)

[Figure 2: System Overview Use Case 24](#_Toc421361193)

[Figure 3: Guest Overview Use Case 25](#_Toc421361194)

[Figure 4: <Guest> Login 25](#_Toc421361195)

[Figure 5: <User> Overview Use Case 27](#_Toc421361196)

[Figure 6: <User> View Report History 27](#_Toc421361197)

[Figure 7: <User> View Room Map 29](#_Toc421361198)

[Figure 8: <User> Create Report 31](#_Toc421361199)

[Figure 9: <Staff> Use case Overview 34](#_Toc421361200)

[Figure 10: <Staff> Create Room Type 35](#_Toc421361201)

[Figure 11: <Staff> Update Room Type 38](#_Toc421361202)

[Figure 12: <Staff> Remove Room Type 41](#_Toc421361203)

[Figure 13: <Staff> Create Classroom 43](#_Toc421361204)

[Figure 14: <Staff> Update Classroom 45](#_Toc421361205)

[Figure 15: <Staff> Remove Classroom 47](#_Toc421361206)

[Figure 16: <Staff> Resolve Report 48](#_Toc421361207)

[Figure 17: <Staff> View Statistic 53](#_Toc421361208)

[Figure 18: <Staff> Mapping Schedule Manual 54](#_Toc421361209)

[Figure 19: <Staff> Mapping Schedule Import File 56](#_Toc421361210)

[Figure 20: <Authorized User> Logout 58](#_Toc421361211)

[Figure 21: <Authorized User> View Profile 59](#_Toc421361212)

[Figure 22: <Authorized User> Update Account 61](#_Toc421361213)

[Figure 23: <Administrator> Add Account 63](#_Toc421361214)

[Figure 24: <Administrator> Activate/Deactivate Account 65](#_Toc421361215)

[Figure 25: <Administrator> Update Account 67](#_Toc421361216)

[Figure 26: <Administrator> Remove Account 69](#_Toc421361217)

[Figure 27: <System> Use case Overview 71](#_Toc421361218)

[Figure 28: <System> Notify about fixed 72](#_Toc421361219)

[Figure 29: <System> Send SMS 73](#_Toc421361220)

[Figure 30: <System> Suggest Available Room 74](#_Toc421361221)

[Figure 31: <System> Notify about damaged 75](#_Toc421361222)

[Figure 32: Conceptual Diagram 78](#_Toc421361223)

# Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Name | Definitions |
| ECRM | Equipment’s Classroom Management |
| HTTP | Hyper Text Transfer Protocol |
| App | Application |
| API | Application Program Interface |
| SMS | Short Message Service |

1. **Introduction**
   1. **Project Information**

* Project Name**: Equipment’s Classroom Management.**
* Project Code**: ECRM.**
* Product Type**: Web Application, Mobile Application.**
* Start Date**: May 11, 2015.**
* End Date**:** 
  1. **Introduction**

Nowadays, numbers of equipment in school are increasing with the rising of students and makes equipment management becoming more important. Therefore, our project looks to meet the demand of managing equipment. We provide website and mobile application for staff to manage equipment of school. In the other hand, teacher can send report about damage equipment in their classroom, staff receives notification about damaged equipment, system automatic analyze damaged level to make recommendations, find available classroom and send SMS for related persons, view equipment statistics.

* 1. **Current Situation**

At the moment, all the equipment in the school is not managed by any software. Staff has to gather much information such as number of classroom, status of classroom, status of equipment in each room…., each 6 or 12 months they must check all the equipment about the damaged. All information had saved in excel files. When there is any problem occurred, teacher has to report directly to staff and it causes wasting time to report. After that, staff will track status, notice to fixer to fix this. In the other hand, staff must check which room is available if changing is needed.

* 1. **Problem Definition**

Advantages:

* Requires less human resources.

Disadvantages:

* Wasting time to report problem.
* Hard to manage equipment by document, book…
* Hard to find available room when needed.
  1. **Proposed Solution**

Building the application supports the staff manage the equipment’s classroom in school. The classroom type is present with position of equipment. Teacher can use web application or mobile application to do check schedule in each day. They can report about damage equipment when teaching or after that by choose equipment, fill information about damaged. After report sent, system will automatic analyze damaged level, check schedule of classroom and automatic change room if needed. The staff also receives notification about damaged.

* + 1. **Feature Function**

Staff can:

* Manage room type: Staff can create and mapping room type with each classroom in school.
* Manage equipment in school: Staff can manage equipment using web application.
* Create schedule by import schedule excel file or create manual.
* Statistic: Staff can see statistic about equipment of school.
* Manage current status of classroom: Staff easy to check status of classroom.
* Resolve report and notice to reporter: Staff can receive and resolve report about damage equipment.
* Change room when needed: Staff can change room or change schedule of teacher when needed.

Admin can:

* Manage account: Admin can create, manage all account of system
* Manage configuration of system.

Teacher can:

* Create report: Teacher can create report about damage equipment
* View schedule: Teacher can check schedule in each day.
* Receive notify about fixed or changing room: Teacher can receive notification about change room if damaged level of room is high.

System can:

* Find available classroom and suggest to staff: System will automatic find available classroom with same options with current class and suggest to staff when needed.
* Send SMS: System will automatic send SMS message to user when needed such as change room, send report…
* System will automatic analyze damaged level and give recommendations
* System will automatic count time remains of equipment and send notification to staff about maintain.
* System will automatic check status of classroom and change schedule of teacher if needed.
  + 1. **Advantages and disadvantages**
* Advantages
* The report is created easier and faster. Saving time for teacher.
* Easy to find the available room if changing room is needed.
* The notifications are received real-time, so the staff can easy to check or resolve.
* Easy to make decision thanks to recommendations from system.
* Staff easily finds the position of failure equipment which shows in classroom map.
* Equipment histories are researched exactly and quickly. (?)
* Show statistics about equipment’s status, using time and changing room in each month…
* All report will be automatic analyze by system.
* Teacher can report when internet connection breakdown.
* Requires less (Staff không cần làm nhiều việc)
* Disadvantages
* The system would not check the behavior of user, so if the report is not correct, the notification will be wrong.
* System can’t apply to complicated classroom likes discuss room…
* The system could not check validate data imported from excel file.
* Staff must have internet connection to use this application.
  1. **Functional Requirement**

Function requirements of the system are listed as below:

* + 1. **Equipment Management**
* Support to manage all equipment in school.
* Track status equipment.
  + 1. **Notification**
* Report the damage about equipment by checking. This also send notify to staff.
* Check the notification and fixing the equipment.
* Notify to reporter about that fix.
* Send SMS.
  + 1. **Room Type Management**
* Support staff to manage all type of room in school.
* Staff can create type of room and map with class in school.
  + 1. **Classroom Management**
* Support to manage all classrooms in school.
* Track status of classroom.
  + 1. **Damaged Analysis And Recommendations**
* Automatic analyze damaged level of classroom and give recommendations.
  + 1. **Suggest Available Classroom**
* Find a similar available classroom with the current classroom.
  + 1. **Manage Account**
* Admin can manage user’s account
* Admin is the person creates all account of system.
  + 1. **Manage Equipment Category**
* Staff can manage equipment category of school.
  + 1. **Tracking Schedule**
* Staff can create schedule manual or import excel file with template.
* Staff can tracking schedule of teacher in each day
* Teacher can see their schedule in each day.
  1. **Role and Responsibility**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Kiều Trọng Khánh | Project Managers | Instructor | khanhkt@fpt.edu.vn |
| 2 | Trần Vĩnh Quang | Developer | Leader | quangtvse61078@fpt.edu.vn |
| 3 | Tăng Việt Hưng | Developer | Member | hungtvse61019@fpt.edu.vn |
| 4 | Đoàn Nguyễn Minh Chí | Developer | Member | chidnmse60717@fpt.edu.vn |
|  |  |  |  |  |

Table : Role and Responsibility

1. **Software Project Management Plan**
   1. **Problem Definition**
      1. **Name of this Capstone Project**

* **Official Name:** Equipment’s Classroom Management.
* **Vietnamese Name:** Ứng dụng hỗ trợ quản lí trang thiết bị cho các phòng học.
* **Abbreviation:** ECRM.
  + 1. **Problem Abstract**

Reporting damaged equipment in classroom is very important. It affects the quality of teaching badly since it caused wasting time. If we can optimize it, the quality of teaching will improve. So the ECRM system will provide the platform where people can reduce reporting damaged equipment time, manage equipment and receive suggestions via web site and mobile application.

* + 1. **Project Overview**
       1. **Current Situation**

The ECRM is the system that helps manages equipment in classroom more convenient way. With teacher, they just click (with web app version) or touch (with mobile app version) on the equipment which is damaged and system will notify to the staff immediately. With staff, they can see specific equipment was damaged in which classroom and who reported it. They can also check which classroom is available due to the schedule excel file imported to the system. Staff can read the statistic about equipment in school if they want to.

The system also has restrictions and it comes from the data in schedule excel file. If staff imports incorrect data but match the template, the result will be wrong. One more thing is the system couldn’t check user’s behaviors. So if teacher report wrong position of the damaged equipment, there are no way the system can deal with it.

* + - 1. **The proposed system**

Our system includes three main subsystems: an online website application for teacher and staff, a mobile application for teacher and a mobile application for staff.

* + - * 1. **Website application**
* **Classroom management:**
* Staff can manage classroom of school.
* Staff can apply each classroom with the specific room type.
* **Room type management:**
  + Staff can manage room type of school.
* **Account management:**
  + Admin can manage all account of system.
  + Admin can activate or deactivate specific account.
* **Report management:**
* Staff can receive report about the damaged equipment from teacher.
* Teacher can send report about damaged equipment when they are teaching and after teach in class.
* **Equipment management:**
* Staff can manage equipment.
* Staff can manage equipment category in school.
* Staff can tracking time remain of specific equipment.
* **Tracking schedule:**
* Staff can create manual schedule for teacher or import schedule by excel file.
* Staff can track schedule of all teacher in system.
* Teacher can see schedule in each day.
* **Statistic:**
* Staff can read statistic about the equipment likes life time, current status…
* **Analysis:**
  + System automatic analyzes the damaged level and gives suggestions.
* **Send SMS:**
  + System sends SMS to related peoples.
* **Suggest available classroom:**
  + System find similar available classroom base on some conditional and give suggestion.
    - * 1. **Mobile application for teacher (ECRM Teacher)**
* Teacher sends report about damaged equipment.
* Teacher sees the map of room with interactive graphic in their account.
* Teacher sees the schedule in current date.
* Teacher receives notification and SMS about changing room if needed.
* Teacher can edit or remove report.
  + - * 1. **Mobile application for staff (ECRM Staff)**
* Staff can get notification from damaged equipment which reported by teacher.
* Staff can send notification to teacher about fixing equipment of changing room if needed
* Staff can send SMS to relative people such as: teacher, security…
* Staff can resolve report when needed.
* Staff can change room when system suggests room fail (?).
  + - 1. **Boundaries of the System**
* The ECRM is used by teacher and staff, run in laptop, PC and android smart phone.
* Language: Vietnamese.
* The lasted product contain:
* The website application for staff and teacher.
* The android application for staff (ECRM Staff).
* The android application for teacher (ECRM Teacher).
  + - 1. **Development Environment**
         1. **Hardware requirements**

**For system**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Operating System | Window Sever 2008 | Window Server 2012 |
| Computer Processor | 512MB RAM | 2GB RAM or more |
| Computer Memory | Intel**®** Core 2 duo | Intel**®** Core™ i5 CPU, M460 @2.53 GHz |
| Internet Connection | Cable, WIFI (2 Mbps) | Cable, WIFI (4Mbps) |

Table : Hardware Requirement for system

* + - * 1. **Software requirements**
* Window Server 2008: Operating system for deploy web service.
* MySQL 5.6: used to create and manage the database for system.
* StarUML v5.0: used to created models and diagrams.
* Skype 7.0: used for communication and meeting.
* IntelliJ IDEA 14.0.3, Android Studio, JDK 7, Apache Tomcat 7, Apache Maven, Android SDK 14: used to implement web application, web service, and mobile application.
* Github & TortoiseSVN 1.8: used for source control.
  1. **Project organization**
     1. **Software Process Model**

The model for project is: Iterative Development Model



Figure : Iterative Development Model

Source: <http://en.wikipedia.org/wiki/Iterative_and_incremental_development>

The reasons for choose iterative development are:

* Members are active but lack of experience so we need to receive feedback during project evolving.
* We have 14 weeks for this project. So we can defined this is small project. Iterative model is suitable for this project to approaching the user thinking.
* This project does not exist, so we must take the survey to customer to know the equipment management in each school.
* In each phase, members only focus for their function. This will make the result better.
* Customer is more actively involved, get higher priority.
* Requirement changed frequently.
  + 1. **Roles and responsibilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Kieu Trong Khanh | Project manager | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | Tran Vinh Quang | Team Leader, BA, Developer, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| **3** | Tang Viet Hung | Team Member, Developer, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| **4** | Doan Nguyen Minh Chi | Team Member, Developer, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |

Table : Roles and Responsibilities Details

* + 1. **Tools and Techniques**
* Front-end: HTML 5, Bootstrap, CSS3, JavaScript, jQuery.
* Back-end: RESTful Web service, Spring MVC, Hibernate, JSP.
* Web-server: Apache Tomcat 7.0.
* Development Tools: IntelliJ IDEA 14
* Database Management System: MySQL 5.6
  1. **Project Management Plan**
     1. **Software development life cycle**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Deliverables** | **Resource needed** | **Dependencies and Constrains** | **Risk** |
| **Room Type Management** | * Support to create, edit room type. | * Website application allowed creating, editing room type of school. * Related document (SRS, SDD, User Guide…) |  | N/A | * Lack of experience. * Technology is difficult. * Not have a clear understanding about business process. |
| **Classroom Management** | * Support to create, edit classroom with room type. * Support to import, mapping schedule. * Support to remove, update classroom. | * Website application allowed creating, editing, removing classroom. * Related document (SRS, SDD, User Guide…) |  | Dependence on “Room Type Management” | * Lack of experience. * Technology is difficult. * Not have a clear understanding about business process. |
| **Equipment**  **Management** | * Support to manage equipment on school. * Support to check time remain, position in classroom… of equipment. * Support to manage equipment category on school. * Support to mapping equipment to each classroom on school. | * Website application allowed managing equipment. * Related document (SRS, SDD, User Guide…) |  | Dependence on “Classroom Management” | * Lack of experience. * Technology is difficult. * Not have a clear understanding about business process. |
| **Account Management** | * Support admin to create, update or remove account from system * Support admin to activate or deactivate account from system | * Website application allows manage account. * Related document (SRS, SDD, User Guide…) |  | N/A | * Lack of experience. * Technology is difficult. * Not have a clear understanding about business process. |
| **Tracking Schedule** | * Support staff to create schedule manual or import excel file * Support staff to check schedule of user * Support user to view schedule in each day. | * Website application allow import schedule, check schedule. * Related document (SRS, SDD, User Guide…) |  | Dependence on “Account Management”, “Classroom Management”. | * Lack of experience. * Technology is difficult. * Not have a clear understanding about business process. |
| **Report Management** | * Support user to create report about damaged equipment. * Support staff to notify about fixing. * Support user to edit or remove report. | * Website application allowed user create and receive report. * Related document (SRS, SDD, User Guide…) |  | Dependence on “Classroom Management”, “Equipment Management” | * Lack of experience. * Technology is difficult. * Not have a clear understanding about business process. |
| **Web Service** | * Build web service to develop android application. | * Web service provides API. * Related document (SRS, SDD, User Guide…) |  | Dependence on “Notify damage and fixing” | * Lack of experience. * Technology is difficult. * Not have a clear understanding about business process. |
| **Android Application** | * Support user to notify about damaged equipment and receive notification in their smartphone using Android OS with internet connection. | * Android application allow user send notify about damage. * Related document (SRS, SDD, User Guide…) |  | Dependence on “Web Service” | * Lack of experience. * Technology is difficult. * Not have a clear understanding about business process. |

Table : Software development life cycle

* + 1. **Phase Detail**
       1. **Phase 1: Room Type Management.**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **Planning** | * Identify start date, end date of iteration. * Identify what team should implement in this iteration. | Tran Vinh Quang |
| **Requirements** | * Collect requirements from customer * Identify and clarify requirements | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Analysis & Design** | * Research and identify technology to implement function * Identify how to implements function * Create ER diagram, database * Create detail design for this function | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Implementation** | * Implement function base on detail design, technology had research. | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Deployment** | * Create deployment documents, user guide. * Deploy this function to server | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Testing** | * Write and run test case for this function * Do unit test and integration test. * Create testing document | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Evaluation** | * Receive feedback | Tran Vinh Quang |

Table : Phase 1: Room Type Management

* + - 1. **Phase 2: Classroom Management.**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **Planning** | * Identify start date, end date of iteration. * Identify what team should implement in this iteration. * Notice about feed of customer about previous iteration to team member. | Tran Vinh Quang |
| **Requirements** | * Collect requirements from customer * Identify and clarify requirements | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Analysis & Design** | * Research and identify technology to implement function * Identify how to implements function * Create, update ER diagram, database * Create detail design for this function | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Implementation** | * Implement function base on detail design, technology had research. | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Deployment** | * Create deployment documents, user guide. * Deploy this function to server | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Testing** | * Write and run test case for this function * Do unit test and integration test. * Update testing document | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Evaluation** | * Receive feedback | Tran Vinh Quang |

Table : Phase 2: Classroom Management

* + - 1. **Phase 3: Equipment Management.**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **Planning** | * Identify start date, end date of iteration. * Identify what team should implement in this iteration. * Notice about feed of customer about previous iteration to team member. | Tran Vinh Quang |
| **Requirements** | * Collect requirements from customer * Identify and clarify requirements | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Analysis & Design** | * Research and identify technology to implement function * Identify how to implements function * Create, update ER diagram, database * Create detail design for this function | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Implementation** | * Implement function base on detail design, technology had research. | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Deployment** | * Create deployment documents, user guide. * Deploy this function to server | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Testing** | * Write and run test case for this function * Do unit test and integration test. * Update testing document | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Evaluation** | * Receive feedback | Tran Vinh Quang |

Table : Phase 3: Equipment Management

* + - 1. **Phase 4: Account Management.**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **Planning** | * Identify start date, end date of iteration. * Identify what team should implement in this iteration. * Notice about feed of customer about previous iteration to team member. | Tran Vinh Quang |
| **Requirements** | * Collect requirements from customer * Identify and clarify requirements | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Analysis & Design** | * Research and identify technology to implement function * Identify how to implements function * Create, update ER diagram, database * Create detail design for this function | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Implementation** | * Implement function base on detail design, technology had research. | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Deployment** | * Create deployment documents, user guide. * Deploy this function to server | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Testing** | * Write and run test case for this function * Do unit test and integration test. * Update testing document | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Evaluation** | * Receive feedback | Tran Vinh Quang |

Table : Phase4: Account Management

* + - 1. **Phase 5: Tracking Schedule.**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **Planning** | * Identify start date, end date of iteration. * Identify what team should implement in this iteration. * Notice about feed of customer about previous iteration to team member. | Tran Vinh Quang |
| **Requirements** | * Collect requirements from customer * Identify and clarify requirements | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Analysis & Design** | * Research and identify technology to implement function * Identify how to implements function * Create, update ER diagram, database * Create detail design for this function | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Implementation** | * Implement function base on detail design, technology had research. | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Deployment** | * Create deployment documents, user guide. * Deploy this function to server | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Testing** | * Write and run test case for this function * Do unit test and integration test. * Update testing document | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Evaluation** | * Receive feedback | Tran Vinh Quang |

Table : Phase 5: Tracking Schedule.

* + - 1. **Phase 6: Report Management**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **Planning** | * Identify start date, end date of iteration. * Identify what team should implement in this iteration. * Notice about feed of customer about previous iteration to team member. | Tran Vinh Quang |
| **Requirements** | * Collect requirements from customer * Identify and clarify requirements | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Analysis & Design** | * Research and identify technology to implement function * Identify how to implements function * Create, update ER diagram, database * Create detail design for this function | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Implementation** | * Implement function base on detail design, technology had research. | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Deployment** | * Create deployment documents, user guide. * Deploy this function to server | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Testing** | * Write and run test case for this function * Do unit test and integration test. * Update testing document | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Evaluation** | * Receive feedback | Tran Vinh Quang |

Table : Phase 6: Report Management.

* + - 1. **Phase 7: Web Service**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **Planning** | * Identify start date, end date of iteration. * Identify what team should implement in this iteration. * Notice about feed of customer about previous iteration to team member. | Tran Vinh Quang |
| **Requirements** | * Collect requirements from customer * Identify and clarify requirements | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Analysis & Design** | * Research and identify technology to implement function * Identify how to implements function * Create, update ER diagram, database * Create detail design for this function | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Implementation** | * Implement function base on detail design, technology had research. | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Deployment** | * Create deployment documents, user guide. * Deploy this function to server | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Testing** | * Write and run test case for this function * Do unit test and integration test. * Update testing document | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Evaluation** | * Receive feedback | Tran Vinh Quang |

Table : Phase 7: Web Service

* + - 1. **Phase 8: Android Application**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Author** |
| **Planning** | * Identify start date, end date of iteration. * Identify what team should implement in this iteration. * Notice about feed of customer about previous iteration to team member. | Tran Vinh Quang |
| **Requirements** | * Collect requirements from customer * Identify and clarify requirements | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Analysis & Design** | * Research and identify technology to implement function * Identify how to implements function * Create, update ER diagram, database * Create detail design for this function | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Implementation** | * Implement function base on detail design, technology had research. | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Deployment** | * Create deployment documents, user guide. * Deploy this function to server | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Testing** | * Write and run test case for this function * Do unit test and integration test. * Update testing document | Tran Vinh Quang, Tang Viet Hung, Doan Nguyen Minh Chi |
| **Evaluation** | * Receive feedback | Tran Vinh Quang |

Table : Phase 8: Android Application

* + 1. **All Meeting Minutes**

Refer to Meeting Minutes folder

<https://github.com/tranquang9a1/ECRM>

* 1. **Coding Convention**

Java: Using to develop desktop application.

Summary:

* Naming Convention.
  + Use camel case for both variable and function name.
  + Use Pascal case for class, interface name.
  + The names of variables declared constants should be all uppercase with words separated by under-scores (“\_”).
* Four spaces should be used as the unit of indentation. The exact construction of the indentation (spaces vs. tabs) is unspecified. Tabs must be set exactly every 8 spaces (not 4).
* When an expression will not fit on a single line, break it according to these general principles:
  + Break after a comma.
  + Break before an operator.
  + Align the new line with the beginning of the expression at the same level on the previous line.
* Declaration.
  + One declaration per line is recommended slice it encourages commenting.
  + In absolutely no case should variables and functions be declared on the same line
  + Do not put different types on the same line.
* Code Examples

Follow “Code Conventions for the Java TM Programming Language, by Sun Microsystems, rev April 20, 1999”.

<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>.

1. **Software Requirement Specification**
   1. **User Requirement Specification**
      1. **Guest Requirement**

Guest is a person who does not login to the system. Guest can use some function in the system. These are some functions guest can use:

* Login
  + 1. **User Requirement**

User is guest, who uses his account to login to the system. In this system, we can define user is a teacher. Member can use some additional function, such as:

* View Report History
* View Room Map
* Create Report
* View Schedule
  + 1. **Staff Requirement**

Staff is manager of the system, we can define staff is an equipment manager in school. Staff can use these function:

* Manage Room Type include:
  + Create New Room Type.
  + Update Room Type.
  + Delete Room Type.
* Manage Classroom include:
  + Create Classroom.
  + Update Classroom.
  + Remove Classroom.
* Manage Schedule include:
  + Map Schedule by Import File.
  + Map Schedule by Manual.
* Statistic.
* Configuration schedule.
* Manage Report include:
  + Resolve Report.
  + Remove Report.
  + Changing Room
* Manage Equipment include:
  + Create Equipment.
  + Remove Equipment.
  + Update Equipment.
  + Map equipment to classroom.
* Manage Equipment Category include:
  + Create equipment category.
  + Remove equipment category.
  + Update equipment category.
    1. **Administrator Requirement**

Admin is the person who manages the system. Admin is super user can use following functions:

* Manage Account include:
  + Add New Account
  + Update Account
  + Remove Account
  + Activate/Deactivate Account
* Manage configuration
  + 1. **Authorize User Requirement**

Authorize User is the person who login to system include user, staff, administrator. Authorize user can use following functions:

* Logout.
* Update Profile.
  1. **System Requirement Specification**
     1. **External Interface Requirement**
        1. **User Interface**
* The user interface uses language Vietnamese
* The user interface display best on 1024x768-screen size for web application and 4-inches for mobile application.
  + - 1. **Hardware Interface**
* N/A
  + - 1. **Software Interface**
* Web Apllication: Firefox (v30 or above), Chrome (v14 or above) browser, enable JavaScript
* Mobile Application: Smartphone with Android operating system (v4.0 or above).
  + - 1. **Communication Protocol**
* Website using HTTP protocol for communication between the web browser and the web server.
* Mobile app using HTTP protocol for communication between app and web service.
  + 1. **System Overview Use Case**

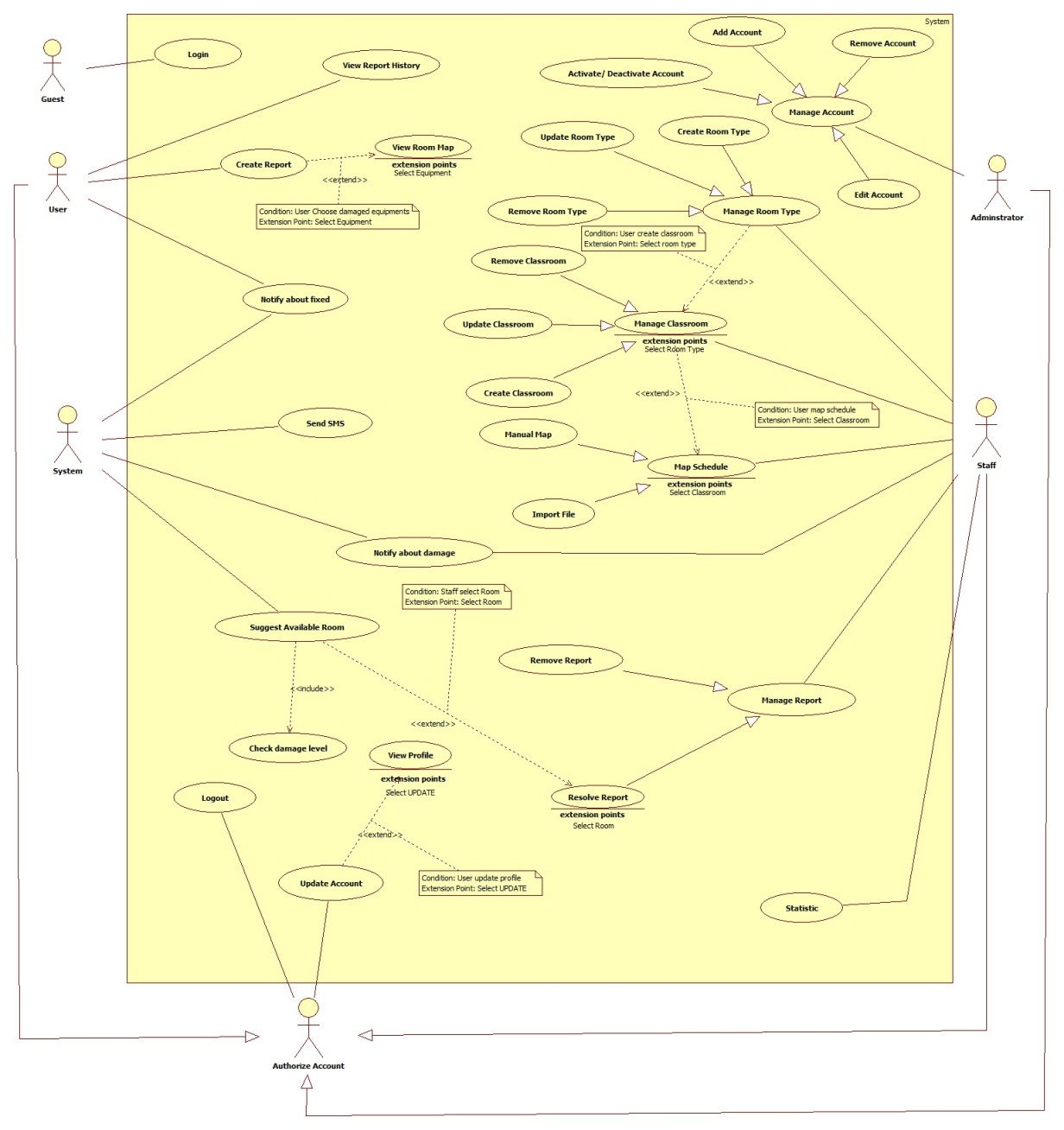


Figure : System Overview Use Case

* + 1. **List of Use Case**
       1. **<Guest> Overview Use Case**

****

Figure : Guest Overview Use Case

* + - 1. **<Guest> Login**



Figure : <Guest> Login

Use case Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM001** | |  |  |
| **Use Case No.** | ECRM001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Login |  |  |
| **Author** | Tang Viet Hung |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:** Guest | |  |  |
| **Summary:** This use case allows guest to login.  **Goal:** Authentication and authorization  **Triggers:** Guest send “Đăng nhập” request to system.  **Preconditions:**   * N/A   **Post Conditions:**   * **Success:** Guest will be logged in system with their roles. Show welcome message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest send Login request. | System navigates to “Đăng nhập” page.   * Username: free text input, min length: 6, max length: 20 required. * Password: free text input, min length: 6, max length: 20, required. * Đăng nhập: command | | 2 | Input field. |  | | 3 | Guest send “Đăng nhập” request. | Guest will login system with their roles.  [Exception 1,2,3] |     **Alternative Scenario:** N/A    **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Wrong username or password. | Show unsuccessful message: “Sai username hoặc password. Hãy thử lại!” | | 2 | “Username” field or “Password” field is empty | Show error message: “Bạn chưa nhập Username or Password. Hãy thử lại” | | 3 | “Username” or “Password” field data length is longer than 20 character or lower than 6 character | Show error message: “Username và Password phải lớn hơn 6 và nhỏ hơn 20 ký tự!” |     **Relationships:** N/A  **Business Rules:**   * After login successfully, guest will login with their role: user, staff and administrator. * Only active user can login to system. * If it is first time user login to system. They will be redirect to “Change Password” page to change password. * On mobile application, user can only login ECRM User App. Staff can only login ECRM Staff App. * On website application, if guest login with role user will be redirect to homepage. If guest is staff, they will be redirected to admin page. | | | |

Table : <Guest> Login

* + - 1. <User> Overview Use Case

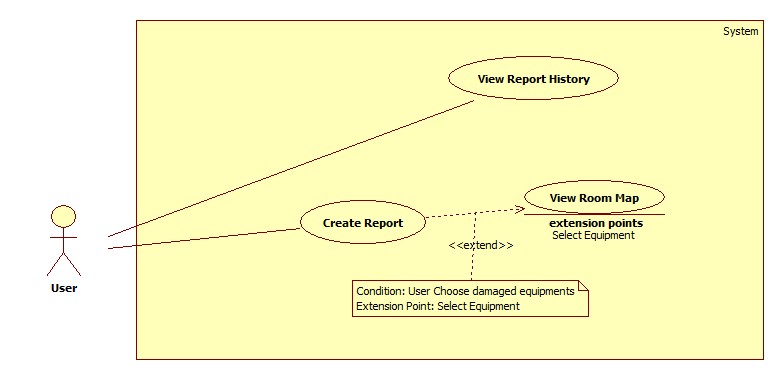


Figure : <User> Overview Use Case

* + - 1. <User> View Report History

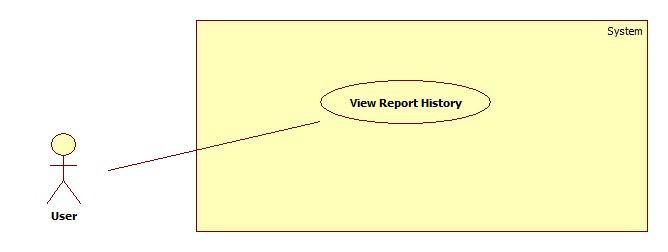


Figure : <User> View Report History

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM002** | | | |
| **Use Case No.** | ECRM002 | **Use Case Version** | 2.0 |
| **Use Case Name** | View Report History | | |
| **Author** | Doan Nguyen Minh Chi | | |
| **Date** | 05/21/2015 | **Priority** | Low |
| **Actor:**   * User   **Summary:**   * This use case allows user to view detail resolve for them report.   **Goal:**   * User can view report detail is resolved.   **Triggers:**   * User send request view report history.   **Preconditions:**   * User must be teacher. * This report must be created by user.   **Post Conditions:**   * **Success:** Show report detail and resolve of staff. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | User send request view report history. |  | |  | System will show report with two tab:   * “Báo cáo” tab:   + “Phòng”: label   + “Người báo cáo”: label   + “Thời gian báo cáo”: label   + “Thiết bị”: label + “Xem bản đồ” send command   + “Mức độ hư hại”: process   + “Mô tả từ giáo viên”: label * “Khắc phục” tab:   + “Nhân viên”: label   + “Thời gian hoàn thành”: label   + “Cách khắc phục”: label   + “Lý do hư hỏng”: lable * “Thoát” send command   [Exception 1][Exception 2] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Report history not exist |  | |  | Show warning message: “Báo cáo hiện không tìm thấy.” | | 2 | User view report of other user |  | |  | Show warning message: “Bạn không có quyền truy cập báo cáo này.” |   **Relationships:** N/A  **Business Rules:**   * User only views them report. * User can view report, if report status is fixed. | | | |

Table : <User> View Report History

* + - 1. <User> View Room Map

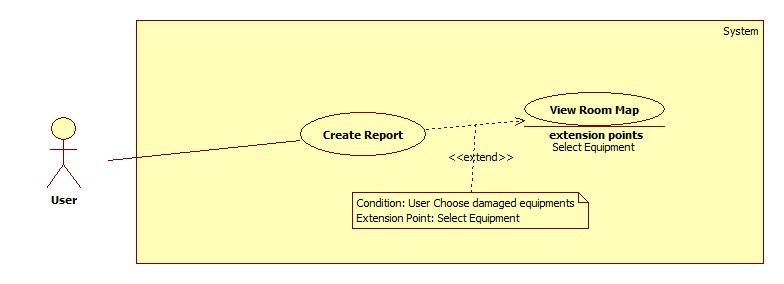
****

Figure : <User> View Room Map

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM003** | |  |  |
| **Use Case No.** | ECRM003 | **Use Case Version** | 2.0 |
| **Use Case Name** | View Room Map |  |  |
| **Author** | Tran Vinh Quang |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:** User | |  |  |
| **Summary:** This use case allows user to view map of classroom they teach.  **Goal:** View map of classroom user teaching. (Apply for web application and mobile application)  **Triggers:** User send “View Room Map” request in Home page.  **Preconditions:**   * User has successfully logged in system. * User must teaching in that time.   **Post Conditions:**   * **Success:** User will see map of classroom they teaching. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User send “View Room Map” request. | System navigates to “Map” page.  It contains the following information   * Map of room they teaching, show location of all equipment in class include board, table, chair, projector, fan…. * Cancel: button   [Exception 1,2] |     **Alternative Scenario:** N/A  **Exceptions:**     |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Connection error | Show error message: “Lỗi kết nối, vui lòng tải lại” | | 2 | User does not teach any class | Show error message: “Bạn không dạy lớp nào! Vui lòng thử lại sau hoặc liên hệ với quản lí”. | | 3 | Classroom did not had type | Show error message: “Hiện tại chưa có sơ đồ phòng học này, vui lòng liên hệ với quản lí”. |   **Relationships:** extend by Create Report  **Business Rules:**   * From map page, user can create report about damage equipment by click to equipment in map. * Only user have teaching in class can see map of room. * If class had damaged equipment, that equipment will be showed with red color. Other equipment will be showed with white color * System will check type of classroom and show map to screen. | | | |

Table : <User> View Room Map

* + - 1. <User> Create Report

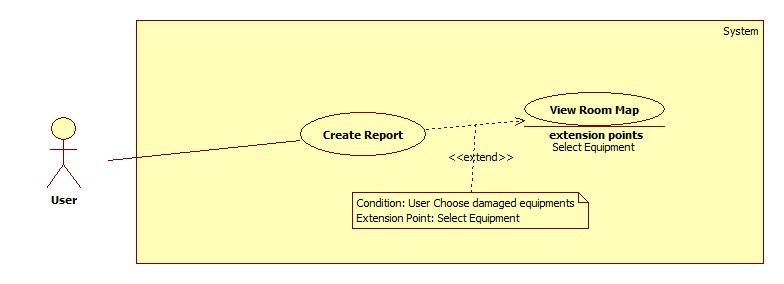


Figure : <User> Create Report

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM004** | | | |
| **Use Case No.** | ECRM004 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create report | | |
| **Author** | Doan Nguyen Minh Chi | | |
| **Date** | 05/21/2015 | **Priority** | High |
| **Actor:**   * User   **Summary:**   * In this use case allows user to send report for staff about damaged equipment.   **Goal:**   * Staff will be received damaged report from user. * Teacher sends report about damaged equipment.   **Triggers:**   * User send “Tạo Báo Cáo” request in notification page.   **Preconditions:**   * User has login successful on system with role teacher. * User has teaching schedule in this classroom.   **Post Conditions:**   * **Success:** Report must be send to staff. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | 1. User sends “Tạo Báo Cáo” request. |  | |  | 1. System will show classroom map which are used by user.    1. Classroom map: a map showing equipment position    2. “Tiếp theo”: send command | | 1. User must choose damaged equipment, and sends “Tiếp theo” request. |  | |  | 1. Detail report will be shown in form:  * “Phòng”: label * “Thiết bị”: label * “Độ hư hỏng”: value list , required   + “Nặng, không thể sử dụng”   + “Trung bình, khó sử dụng”   + “Nhẹ, cố thể sử dụng” * “Đánh giá của bạn”: value list, required   + “Cần đổi phòng ngay”   + “Cần hổ trợ gấp”   + “Có thể sửa sau” * “Mô tả hư hại”: free text input, min length: 0, max length: 200, optional. * “Gửi báo cáo”: send command   [Exception 1, 2] | | 1. User sends “Gửi báo cáo” request with complete information. |  | |  | 1. Report will be saved.   [Exception 3] |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Send “Tiếp theo” request when no equipment was chosen. |  | |  | Show message: “Không có thiết bị nào được chọn”. | | 2 | User chooses the equipment that has already been reported. |  | |  | Show message: “Thiết bị này đã được báo cáo.” | | 3 | Connection error. |  | |  | Show error message: “Lỗi kết nối, vui lòng thử lại” |   **Relationships:** include Notify about damage, extend by View Room Map  **Business Rules:**   * Only user can report equipment in classroom which they are teaching. * If connection is lost, report will be saved in local. Then it will be resent. * System will get value of “Độ hư hỏng”, “Đánh giá của bạn”, and priority of equipment, before calculate damage level of them. It will be notify for staff in resolve form. * If damage level is larger than 35%, system will call Suggest Available Room. | | | |

Table : <User> Create Report

* + - 1. <Staff> Use Case Overview

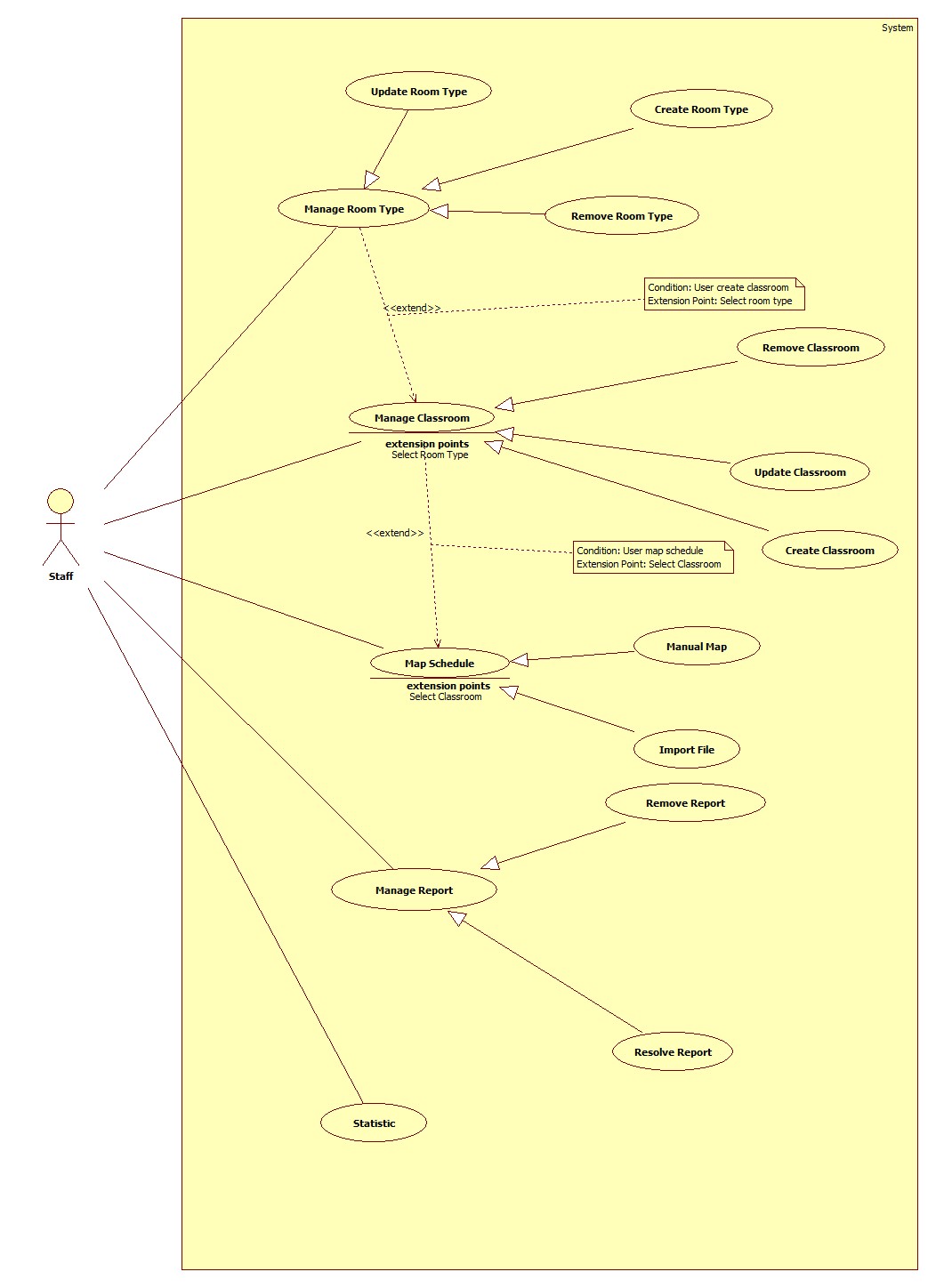


Figure : <Staff> Use case Overview

* + - 1. <Staff> Create New Room Type

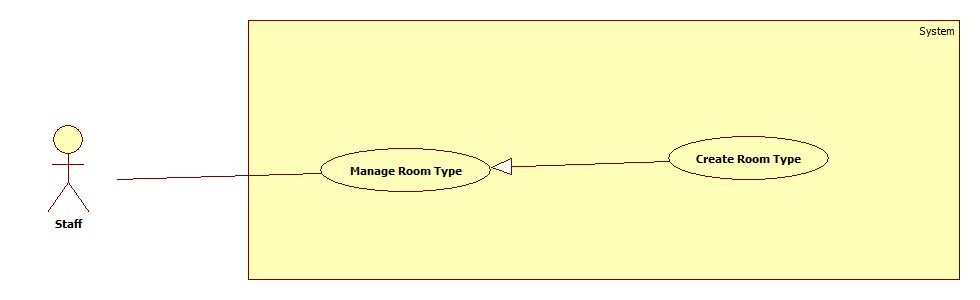


Figure : <Staff> Create Room Type

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM005** | |  |  |
| **Use Case No.** | ECRM005 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create Room Type |  |  |
| **Author** | Tang Viet Hung |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:** Staff | |  |  |
| **Summary:** This use case allows staff to create new room type.  **Goal:** Create room type to the system.  **Triggers:** Staff sends request “Tạo Kiểu Phòng” in “Kiểu Phòng” page.  **Preconditions:**   * User login as staff role.   **Post Conditions:**   * **Success:** New room type will be created and show successful message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff focuses on “Phòng Học” and clicks “Kiểu Phòng”. | System navigates to “Kiểu Phòng” page. Page includes:   * Text box search. * Dropdown list: value list “Số kiểu”, “Số chỗ ngồi”. * Search: command. * Table with columns: * Số kiểu * Số chỗ ngồi * Check box * Xem: command. * Tạo mới: command   [Alternative 1, 2] | | 2 | Staff sends “Tạo Kiểu Phòng” request. | System navigates to “Tạo Mới Kiểu Phòng” page. Page includes:   * Số kiểu: textbox, disabled. * Máy lạnh: check box. * Máy chiếu: check box. * Tivi: check box. * Quạt: check box. * Loa: check box. * Số lượng dãy dọc: text box, number type, required. * Chấp nhận: command. * Cancel: command.   [Exception 1]  [Alternative 4] | | 3 | Staff inputs two textbox and sends “Chấp nhận” request. | System will show table with number of column equal with number staff inputted in “Số lượng dãy dọc”. Each column includes:   * Số dãy ngang: text box, number type, required. * Số chỗ ngồi mỗi dãy ngang: text box, number type, required. * Xem: command.   [Exception 2, 3, 4] | | 4 | Staff inputs all fields and sends “Xem” request. | System navigates to “Xem Kiểu Phòng” Page. Page includes:   * Classroom map. * Tạo Kiểu Phòng: command. * Cancel: command.   [Alternative 5] | | 5 | Staff sends “Tạo Kiểu Phòng” request. | System show successful message: “Tạo kiểu phòng mới thành công!” and navigate to “Kiểu Phòng” page. |     **Alternative Scenario:**     |  |  |  | | --- | --- | --- | | No | Actor Action | System Respnse | | 1 | Staff inputs to text box search and sends “Tìm kiếm” request. | Show result in table. | | 2 | Staff send request “Xem”. | Navigate to “Xem Kiểu Phòng” page. | | 4 | Staff sends “Cancel” request “Tạo Mới Kiểu Phòng” page. | System navigates to “Kiểu Phòng” page. | |  | Staff sends “Cancel” request “Xem Kiểu Phòng” page. | System navigates to “Tạo Mới Kiểu Phòng” page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | “Số lượng dãy dọc” is empty or characters. | Show error message: “Số lượng dãy dọc không hợp lệ!” | | 2 | “Số chỗ ngồi mi dãy ngang” is empty or characters. | Show error message: “Số chỗ ngồi mỗi dãy ngang không hợp lệ!” | | 3 | Staff creates room type that has number of slots is more than 50 slots. | System show room map with only one chair and one table represent for the whole slots. | | 4 | Staff creates room type that has number of slots is 0. | Show error message: “Số lượng chỗ ngồi đủ để tạo kiểu phòng!” |   **Relationships:** N/A  **Business Rules:**   * System only support classroom with less than 50 slots. If it’s more than 50 slots, system will show room map with only one chair and one table represent for the whole slots. * “Số kiểu” is automatically generated by the system. It’s taken from database which is key. | | | |

Table : <Staff> Create Room Type

* + - 1. <Staff> Update Room Type

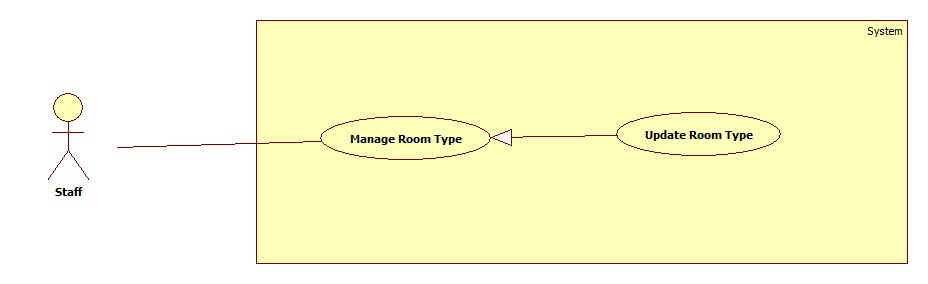


Figure : <Staff> Update Room Type

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM006** | |  |  |
| **Use Case No.** | ECRM006 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update Room Type |  |  |
| **Author** | Tang Viet Hung |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:** Staff | |  |  |
| **Summary:** This use case allows staff to update a room type.  **Goal:** Update a specific room type.  **Triggers:** Staff sends request “Chỉnh Sửa” on “Xem Kiểu Phòng” page.  **Preconditions:**   * User login as staff role. * Room type existed in database.   **Post Conditions:**   * **Success:** Room type will be updated and show successful message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff focuses on “Phòng Học” and clicks “Kiểu Phòng”. | System navigates to “Kiểu Phòng” page. Page includes:   * Text box search. * Dropdown list. * Search: command. * Table with columns: * Số kiểu * Số chỗ ngồi * Check box * Xem: command. * Tạo mới: command   [Alternative 1, 2] | | 2 | Staff sends “Xem” request. | System navigates to “Xem Kiểu Phòng” page. Page includes:   * Sơ đồ kiểu phòng: map. * Xóa: command. * Chỉnh sửa: command. * Quay về: command.   [Alternative 3, 4] | | 3 | Staff sends “Chỉnh sửa” request. | System navigates to “Chỉnh Sửa Kiểu Phòng” page. Page includes:   * Số kiểu: textbox, disabled. * Máy lạnh: check box. * Máy chiếu: check box. * Tivi: check box. * Quạt: check box. * Loa: check box. * Số lượng dãy dọc: text box, number type, required. * Chấp nhận: command. * Cancel: command.   [Exception 1]  [Alternative 5] | | 4 | Staff inputs two textboxs and sends “Chấp nhận” request. | System will show table with number of column equal with number staff inputted in “Số lượng dãy dọc”. Each column includes:   * Số dãy ngang: text box, number type, required. * Số chỗ ngồi mỗi dãy ngang: text box, number type, required. * Xem: command.   [Exception 2, 3, 4] | | 4 | Staff inputs all fields and sends “Xem” request. | System navigates to “Xem Kiểu Phòng” Page. Page includes:   * Classroom map. * Tạo Kiểu Phòng: command. * Cancel: command. * [Alternative 6] | | 5 | Staff sends “Tạo Kiểu Phòng” request. | System show successful message: “Chỉnh sửa kiểu phòng thành công” and navigate to “Kiểu Phòng” page. |     **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff inputs to text box search and sends “Tìm kiếm” request. | Show result in table. | | 2 | Staff sends “Xem” request. | Navigate to “Xem Kiểu Phòng” page. | | 3 | Staff sends “Xóa” request. | Show message: “Bạn có chắc muốn xóa kiểu phòng?” | |  | Staff sends “Quay về” request. | Naviages to “Kiểu Phòng” page. | | 5 | Staff sends “Cancel” request in “Chỉnh Sửa Kiểu Phòng” page. | System navigates to “Kiểu Phòng” page. | | 6 | Staff sends “Cancel” request in “Xem Kiểu Phòng” page. | System navigates to “Chỉnh Sửa Kiểu Phòng” page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | “Số lượng dãy dọc” is empty or characters. | Show error message: “Số lượng dãy dọc không hợp lệ!” | | 2 | “Số chỗ ngồi mỗi dãy ngang” is empty or characters. | Show error message: “Số chỗ ngồi mỗi dãy ngang không hợp lệ!” | | 3 | Staff creates room type that has number of slots is more than 50 slots. | System show room map with only one chair and one table represent for the whole slots. | | 4 | Staff creates room type that has number of slots is 0. | Show error message: “Số lượng chỗ ngồi đủ để tạo kiểu phòng!” |   **Relationships:** N/A  **Business Rules:**   * System will automatically inputs all filed that existed on page “Chỉnh Sửa Kiểu Phòng”. It’s taken from database. * System only support classroom with less than 50 slots. If it’s more than 50 slots, system will show room map with only one chair and one table represent for the whole slots. * “Số kiểu” is automatically generated by the system. It’s taken from database which is key. | | | |

Table : <Staff> Update Room Type

* + - 1. <Staff> Remove Room Type

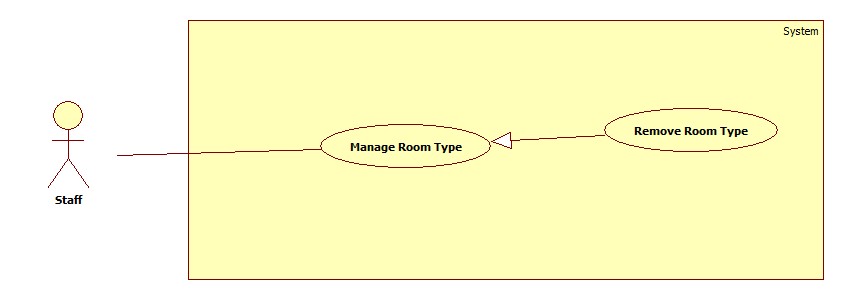


Figure : <Staff> Remove Room Type

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM007** | |  |  |
| **Use Case No.** | ECRM007 | **Use Case Version** | 2.0 |
| **Use Case Name** | Remove Room Type |  |  |
| **Author** | Tang Viet Hung |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:** Staff | |  |  |
| **Summary:** This use case allows staff to delete a room type.  **Goal:** Delete a specific room type.  **Triggers:** Staff sends request “Xóa” in “Xem Kiểu Phòng” page.  **Preconditions:**   * User login as staff role. * Room type existed in database.   **Post Conditions:**   * **Success:** Room type will be deleted and show successful message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff focuses on “Phòng Học” and clicks “Kiểu Phòng”. | System navigates to “Kiểu Phòng” page. Page includes:   * Text box search. * Dropdown list. * Search: command. * Table with columns: * Số kiểu * Số chỗ ngồi * Check box * Xem: command. * Tạo mới: command   [Alternative 1, 2] | | 2 | Staff sends “Xem” request. | System navigates to “Xem Kiểu Phòng” page. Page includes:   * Sơ đồ kiểu phòng: map. * Xóa: button. * Chỉnh sửa: command. * Quay về: command.   [Alternative 3, 4] | | 3 | Staff sends “Xóa” request. | Show dialog: “Bạn có chắc muốn xóa kiểu phòng?”  [Alternative 5] | | 4 | Staff clicks “Yes” button in dialog. | System shows successful message: “Xóa kiểu phòng thành công” and navigates to “Kiểu Phòng” page. |     **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff inputs to text box search and sends “Tìm kiếm” request. | Show result in table. | | 2 | Staff sends “Tạo Mới” request. | Navigate to “Tạo Mới Kiểu Phòng” page. | | 3 | Staff sends “Sửa” request. | Navigate to “Chỉnh Sửa Kiểu Phòng” page. | |  | Staff sends “Quay về” request. | Naviages to “Kiểu Phòng” page | | 5 | Staff clicks “No” in dialog | System remains in “Xem Kiểu Phòng” Page. |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * After receiving request, system will delete room type in database and all the classroom that use this room type will be unable to view classroom map. | | | |

Table : <Staff> Remove Room Type

* + - 1. <Staff> Create Classroom

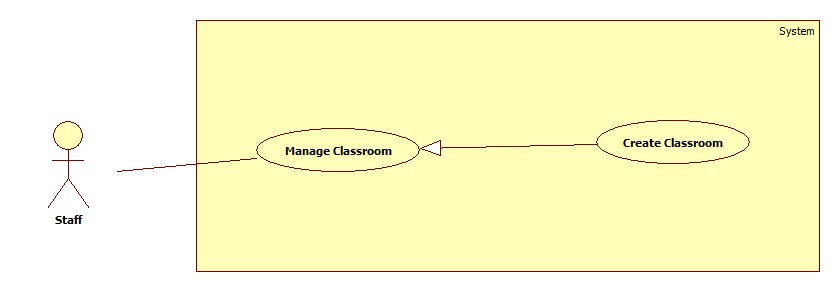


Figure : <Staff> Create Classroom

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM008** | |  |  |
| **Use Case No.** | ECRM008 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create Classroom |  |  |
| **Author** | Tran Vinh Quang |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:**   * Staff | |  |  |
| **Summary:**   * This use case allows staff to create new classroom (applied for web application)   **Goal:** Add new classroom to database  **Triggers:** Staff sends request “Thêm Lớp Học” in classroom management page.  **Preconditions:**   * User must login into the system with staff role.   **Post Conditions:**   * **Success:** New classroom will be added to database. Success message will be shown. * **Fail:** Error message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends request “Thêm Lớp Học” in manage page. | System navigates to “Thêm Lớp Học” page.  Show form with the following information:   * Tên phòng: free text input, min length: 1, max length: 100, required. * Loại phòng: Value of list, required (Phòng 25 ghế, Phòng 30 ghế…) * Tạo mới: command * Hủy: command   [Alternative 1] | | 2 | Staff fills necessary information into the form |  | | 3 | Staff send “Tạo mới” request. | Classroom is added to database, success message “Thêm mới lớp học thành công” is shown.  System is navigation to “Classroom Management Page”.  [Exception 1,2,3] |     **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff click “Hủy” | System navigation to “Classroom Management Page” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | “Tên phòng học” field is empty | Error message “Tên phòng học không được để trống!” is shown. | | 2 | “Tên phòng học” field data length is longer than 100 characters. | Error message “Tên phòng học không được quá 100 ký tự!” is shown. | | 3 | “Tên phòng học” field is duplicated in database | Error message “Phòng học có tên này đã được tạo, vui lòng sử dụng tên khác” is shown. |   **Relationships:** N/A  **Business Rules:**   * Each classroom will have only 1 type. * Button “Tạo mới” will enable when all of information have been filled. | | | |

Table : <Staff> Create Classroom

* + - 1. <Staff> Update Classroom

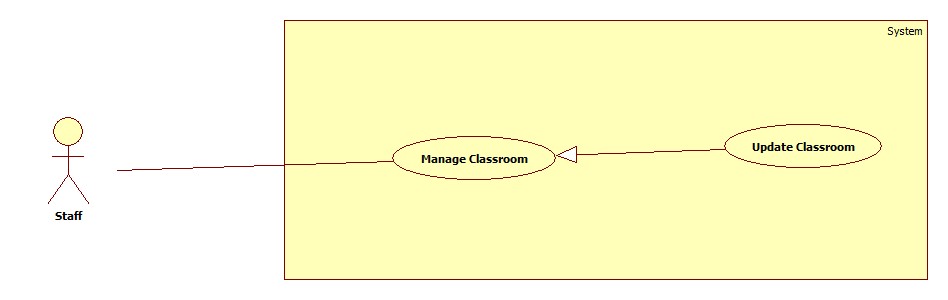


Figure : <Staff> Update Classroom

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM009** | |  |  |
| **Use Case No.** | ECRM009 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update Classroom |  |  |
| **Author** | Tran Vinh Quang |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:**   * Staff | |  |  |
| **Summary:**   * This use case allows staff to update information of classroom (applied for web application)   **Goal:** Update existed classroom.  **Triggers:** Staff sends “Sửa” command in classroom management page.  **Preconditions:**   * User must login into the system with staff role. * Classroom exists in database.   **Post Conditions:**   * **Success:** Classroom will be updated with new information. Success message will be shown. * **Fail:** Classroom will not be updated. Error message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff choose classroom and sends request to update exist classroom. | System navigates to classroom detail page.  Show form with the following information:   * Tên phòng: Text input, min length: 1, max length: 100, required. * Loại phòng: List of value, required (Phòng 25 ghế, Phòng 30 ghế…) * Lưu: command * Hủy: command   [Alternative 1] | | 2 | Staff fills necessary information into the form |  | | 3 | Staff sends “Lưu” command. | Confirm dialog will be shown  [Exception 1,2,3]  [Alternative 1,] | | 4 | Staff click “Đồng ý” button | Classroom will be updated and show successful message. System navigation to Classroom Management Page |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff click “Hủy” | Return to classroom management page. | | 1 | Staff click “Cancel” | Confirm dialog will be canceled and no change is made. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | “Tên phòng học” field is empty | Error message “Tên phòng học không được để trống!” is shown. | | 2 | “Tên phòng học” field data length is longer than 100 characters. | Error message “Tên phòng học không được quá 100 ký tự!” is shown. | | 3 | “Tên phòng học” field is duplicated in database | Error message “Phòng học có tên này đã được tạo, vui long sử dụng tên khác” is shown. |   **Relationships:** N/A  **Business Rules:**   * Staff can update name and type of classroom. | | | |

Table : <Staff> Update Classroom

* + - 1. <Staff> Remove Classroom

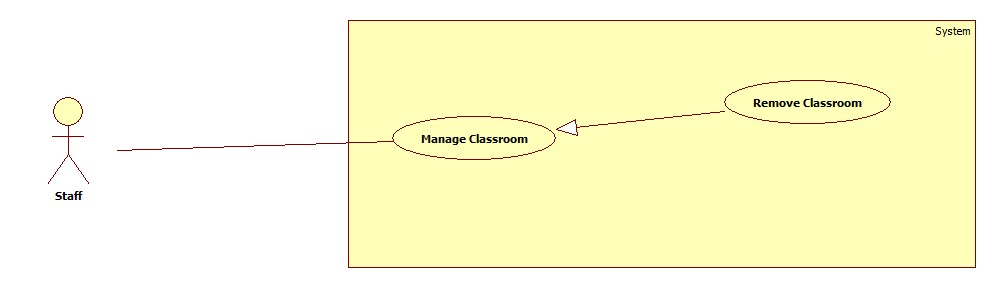


Figure : <Staff> Remove Classroom

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM010** | |  |  |
| **Use Case No.** | ECRM010 | **Use Case Version** | 2.0 |
| **Use Case Name** | Remove Classroom |  |  |
| **Author** | Tran Vinh Quang |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:**   * Staff | |  |  |
| **Summary:**   * This use case allows staff to remove of classroom (applied for web application)   **Goal:** Remove existed classroom from system.  **Triggers:** Staff send “Xóa” request in classroom management page.  **Preconditions:**   * User must login into the system with staff role. * Classroom exists in database.   **Post Conditions:**   * **Success:** Classroom will be deleted from database. Success message will be shown. * **Fail:** Classroom will not be deleted. Error message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff choose classroom and send request to delete classroom | System navigates to classroom detail page.  Show form with the following information:   * Tên phòng: Text input, min length: 1, max length: 100, required. disable * Loại phòng: Value of list, required (Phòng 25 ghế, Phòng 30 ghế…) disable * Xóa: command * Hủy: command   [Alternative 1] | | 2 | Staff send “Xóa” command | Confirm dialog will be shown [Exception 1,2,3,4,5,6,7]  [Alternative 1] | | 3 | Staff click “OK” | Classroom will be deleted and show successful message. System navigation to Classroom Management Page |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Staff click “Cancel” | Confirm dialog will be canceled and no change is made. |   **Exceptions: N/A**  **Relationships:** N/A  **Business Rules:**   * After delete classroom, all of information about class such as damage equipment, type, and schedule will be deleted. | | | |

Table : <Staff> Remove Classroom

* + - 1. <Staff> Resolve Report

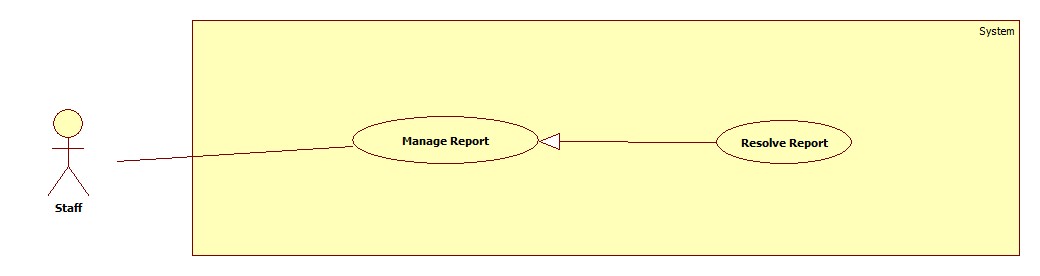


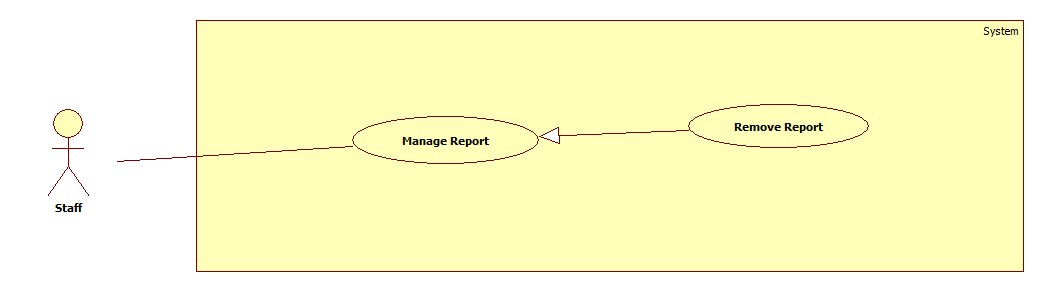
Figure : <Staff> Resolve Report

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM011** | | | |
| **Use Case No.** | ECRM011 | **Use Case Version** | 2.0 |
| **Use Case Name** | Resolve Report | | |
| **Author** | Doan Nguyen Minh Chi | | |
| **Date** | 05/21/2015 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff to resolve damaged equipment in report, or all of damaged equipment in room.   **Goal:**   * Report will be resolved, and user will receive notification.   **Triggers:**   * Staff sends request view new report notification.   **Preconditions:**   * User must be staff.   **Post Conditions:**   * **Success:** Report must be resolved and user receives resolved massage. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | 1. Staff sends request view new notification. |  | |  | 1. System will show report detail of notify with two tab and some button:  * “Báo cáo” tab:   + “Phòng”: label   + “Người báo cáo”: label   + “Thời gian báo cáo”: label   + “Thiết bị”: label + “Vị trí” send command   + “Mức độ hư hại”: value ranges [0-100]   + “Mô tả từ giáo viên”: label   + “Đề nghị đổi phòng”: label + “Đổi phòng” send command * “Lịch sử” tab:   + “Thiết bị”: value list     - “Máy chiếu”     - “Máy lạnh”     - “Máy quạt”     - ……...   + Table history damaged of this equipment * “Khắc phục” send command * “Xóa” send command * “Xóa tất cả” send command * “Khắc phục tất cả” send command * “Thoát” send command | | 1. Staff sends “Khắc phục” request   [Alternative 1] [Alternative 2]  [Alternative 3] [Alternative 4] |  | |  | 1. System saves resolve data.   [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff sends “Xem bản đồ” request |  | |  | Show classroom map with position of damaged equipment | | 2 | 1. Staff sends “Đổi phòng” request |  | |  | 1. Show change room form with:  * “Phòng trống”: label * “Phòng khác”: send command * “Đổi” send command * “Thoát” send command | | 1. Staff sends “Đổi” request   [Alternative 5, 6] |  | |  | 1. Change to new room for class | | 3 | Staff chooses “Lịch sử” |  | |  | Show resolved history of equipment in room | | 4 | 1. Staff sends “Khắc phục tất cả” request |  | |  | 1. Show list all damaged equipment in this room.  * Equipment image: image   + Equipment name: label   + Number of damaged equipment. | | 1. Staff will chooses some damaged equipment |  | |  | 1. System saves resolved data. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Staff sends “Khắc phục” request but the report has already resolved. |  | |  | Show message: “Báo cáo đã được xử lý bởi một nhân viên khác.” | | 2 | System cannot send SMS. |  | |  | System will resent at 10 minute later. |   **Relationships:**  **Business Rules:**   * SMS will be sent two times. * All damaged equipment always update when user report other equipment in this room. * If damaged level of room is higher than 35%, the system will show available rooms. * When staff resolves a report, it notifies for other staff. * Equipment in other report are fixed in report, it will update in other report. * Report’s status will change to “going” when all the damaged equipment was not fixed. A damaged equipment change to “finish” status after it was fixed. * Report’s status will be changed to “finish” when all damaged equipment was fixed. | | | |

Table : <Staff> Resolve Report

* + - 1. **<Staff> Remove Report**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM012** | | | |
| **Use Case No.** | ECRM012 | **Use Case Version** | 2.0 |
| **Use Case Name** | Remove Report | | |
| **Author** | Tran Vinh Quang | | |
| **Date** | 05/21/2015 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case allows staff to remove report   **Goal:**   * Report will be removed   **Triggers:**   * Staff sends request remove report   **Preconditions:**   * User must login to system with role staff.   **Post Conditions:**   * **Success:** Report must be removed. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  | | --- | --- | | Actor Action | System Response | | Staff sends request view new notification. |  | |  | System will show report detail of notify with two tab and some button:   * “Báo cáo” tab:   + “Phòng”: label   + “Người báo cáo”: label   + “Thời gian báo cáo”: label   + “Thiết bị”: label + “Vị trí” send command   + “Mức độ hư hại”: value ranges [0-100]   + “Mô tả từ giáo viên”: label   + “Đề nghị đổi phòng”: label + “Đổi phòng” send command * “Lịch sử” tab:   + “Thiết bị”: value list     - “Máy chiếu”     - “Máy lạnh”     - “Máy quạt”     - ……...   + Table history damaged of this equipment * “Khắc phục” send command * “Xóa” send command. * “Xóa tất cả” send command * “Khắc phục tất cả” send command * “Thoát” send command | | Staff sends “Xóa” request |  | |  | System remove report  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff sends “Xem bản đồ” request |  | |  | Show classroom map with position of damaged equipment | | 2 | Staff chooses “Lịch sử” |  | |  | Show resolved history of equipment in room | | 3 | Staff sends “Xóa tất cả” request |  | |  | Show list all damaged equipment in this room.   * Equipment image: image   + Equipment name: label   + Number of damaged equipment. | | Staff will chooses some damaged equipment |  | |  | System removed report had choose. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Staff sends “Xóa” request but the report has not choose. |  | |  | Show message: “Bạn phải chọn 1 report mới thực hiện chức năng này.” | | 2 | System cannot send SMS. |  | |  | System will resent at 10 minute later. |   **Relationships:**  **Business Rules:**   * SMS will be sent two times. * All damaged equipment always update when user report other equipment in this room. * When staff removes a report, it notifies for other staff. * Report’s status will change to “remove” after complete this function. | | | |

Table : <Staff> Remove Report

* + - 1. <Staff> View Statistic

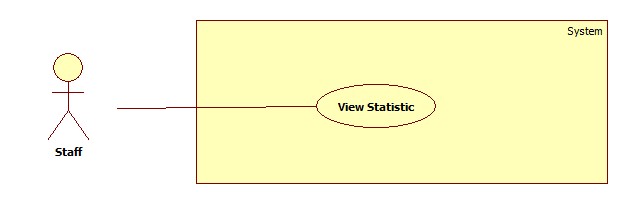


Figure : <Staff> View Statistic

**Use case specification**

* + - 1. <Staff> Mapping Schedule Manual

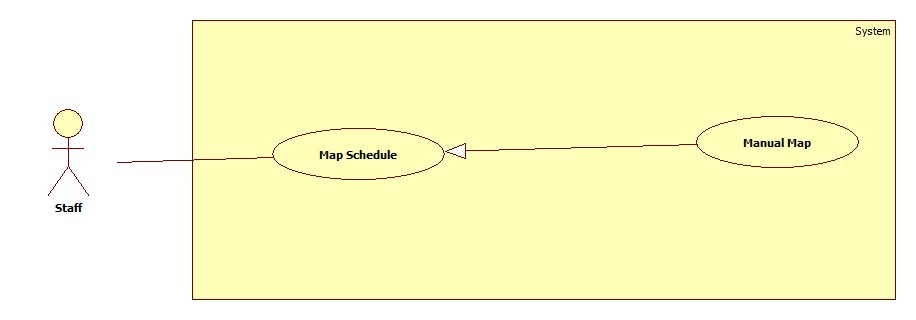


Figure : <Staff> Mapping Schedule Manual

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM014** | |  |  |
| **Use Case No.** | ECRM014 | **Use Case Version** | 2.0 |
| **Use Case Name** | Mapping Schedule Manual |  |  |
| **Author** | Tang Viet Hung |  |  |
| **Date** | 21/05/2015 | **Priority** | Normal |
| **Actor:** Staff | |  |  |
| **Summary:** This use case allows staff to import file schedule.  **Goal:** Import a schedule file.  **Triggers:** Staff sends “Manual” request on “Nhập Lịch” page.  **Preconditions:**   * User login as staff role.   **Post Conditions:**   * **Success:** Schedule file will be imported and show successful message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff focuses on “Lịch” and clicks “Nhập Lịch”. | System navigates to “Nhập Lịch” page. Page includes:   * Nhập tay: command. * Tìm: command. * Tải template: command * Import: command.   [Alternative 1] | | 2 | Staff sends “Nhập tay” request. | System navigates to “Nhập Tay” page. Page includes form:   * Giáo viên: text input, min length: 6, max length: 30, required. * Giờ dạy bắt đầu: text input, time type, required. * Giờ dạy kết thúc: text input, time type, required. * Ngày dạy bắt đầu: text input, date type, required. * Ngày dạy kết thúc: text input, date type, required. * Số lượng học sinh: text input, number type, required. * Tên phòng: dropdown list, required. * Import: command. * Cancel: command.   [Exception 1, 2, 3, 4 ,5, 6] | | 3 | Staff inputs: “Giáo viên”, “Giờ dạy bắt đầu”, “Giờ dạy kết thúc”, “Ngày dạy bắt đầu”, “Ngày dạy kết thúc”, “”Number of student”. | System will add data into “Tên phòng” dropdown list.  [Exception 7, 8] | | 4 | Staff sends “Import” request.  [Alternative 1] | System shows successful message. |     **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff sends “Cancel” request. | System navigates to “Nhập Lịch” page |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | “Giáo viên” text input is empty or not in range. | System shows error message: “Giáo viên phải từ 6 tới 30 ký tự. Vui lòng thử lại!” | | 2 | “Giờ dạy bắt đầu” text input is empty or not time type. | System shows error message: “Giờ dạy bắt đầu không hợp lệ. Vui lòng thử lại!” | | 3 | “Giờ dạy kết thúc” text input is empty or not time type. | System shows error message: “Giờ dạy kết thúc không hợp lệ. Vui lòng thử lại!” | | 4 | “Ngày dạy bắt đầu” text input is empty or not time type. | System shows error message: “Ngày dạy bắt đầu không hợp lệ. Vui lòng thử lại!” | | 5 | “Ngày dạy kết thúc” text input is empty or not time type. | System shows error message: “Ngày dạy kết thúc” không hợp lệ. Vui lòng thử lại!” | | 6 | System could not show any data. | System shows error message: “Không tìm thấy phòng học phù hợp. Vui lòng thử lại!” | | 6 | “Tên phòng” is not selected. | System shows error message: “Phải chọn một lớp học. Vui lòng thử lại!” |   **Relationships:** N/A  **Business Rules:**   * **“**Tên phòng**”** will appear after staff inputted all the text input above. * In manual function, after inputting all the field, system will check in database that if any classroom are appropriate with data that staff inputted. System will allow staff to import schedule if and only if there are classroom available. | | | |

Table : <Staff> Mapping Schedule Manual

* + - 1. <Staff> Mapping Schedule Import File

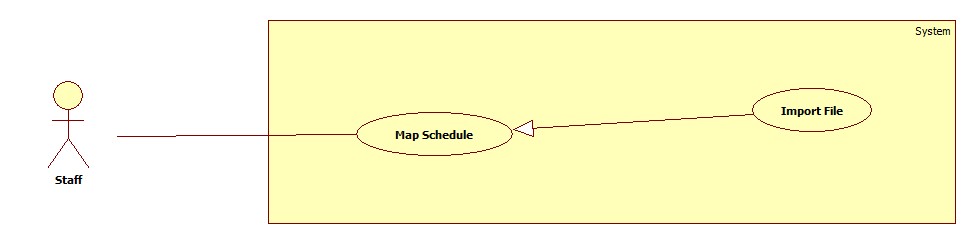


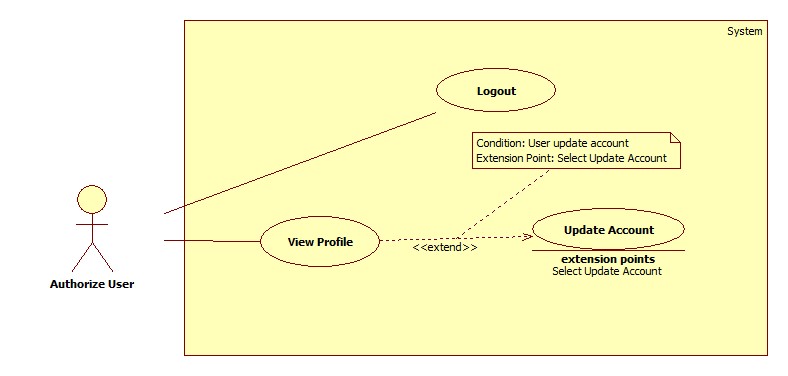
Figure : <Staff> Mapping Schedule Import File

Use case specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM015** | |  |  |
| **Use Case No.** | ECRM015 | **Use Case Version** | 2.0 |
| **Use Case Name** | Import File |  |  |
| **Author** | Tang Viet Hung |  |  |
| **Date** | 21/05/2015 | **Priority** | Normal |
| **Actor:** Staff | |  |  |
| **Summary:** This use case allows staff to import file schedule.  **Goal:** Import a schedule file.  **Triggers:** Staff sends “Import” request on “Nhập Lịch” page.  **Preconditions:**   * User login as staff role.   **Post Conditions:**   * **Success:** Schedule file will be imported and show successful message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff focuses on “Lịch” and clicks “Nhập Lịch”. | System navigates to “Nhập Lịch” page. Page includes:   * Tìm: command. * Tải template: command * Import: command.   [Alternative 1] | | 2 | Staff clicks on “Tìm” button and choose schedule excel file from his computer. | System shows the name of schedule file. | | 4 | Staff clicks “Import” button. | System shows successful message.  [Exception 1] |     **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff sends “Tải template” request. | Staff will get template excel file from server. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff import a wrong template excel file. | System shows error message. |   **Relationships:** N/A  **Business Rules:**   * After receiving request, system will check validate the file that got from user. If it’s incorrect template, system will show error. The new schedule will be saved in database and system will delete the old one. | | | |

Table : <Staff> Mapping Schedule Import File

* + - 1. **<Authorized User> Use case overview**

****

* + - 1. <Authorized User> Logout

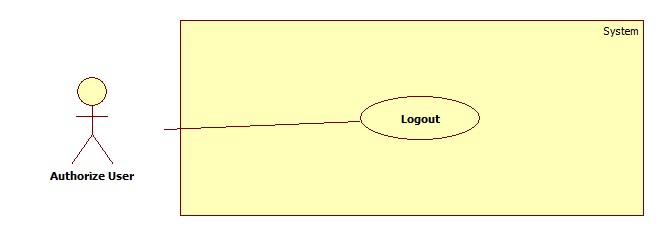


Figure : <Authorized User> Logout

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM016** | |  |  |
| **Use Case No.** | ECRM016 | **Use Case Version** | 2.0 |
| **Use Case Name** | Logout |  |  |
| **Author** | Tang Viet Hung |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:** Guest | |  |  |
| **Summary:** This use case allows user to logout.  **Goal:** Logout of the system  **Triggers:** User click on “Logout” button  **Preconditions:**   * User has successfully logged in system.   **Post Conditions:**   * **Success:** User will be logged out of system. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User send “Đăng xuất” request. | System show dialog: “Bạn có chắc muốn đăng xuất?” | | 2 | User clicks “Yes” button. | System navigates to login screen. |     **Alternative Scenario:**     |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User clicks “No” button in dialog. | System remains user in current page. |   **Exceptions:** N/A    **Relationships:** N/A  **Business Rules:**   * After logout successfully, guest cannot use the system. | | | |

Table : <Authorized User> Logout

* + - 1. **<Authorized User> View Profile**

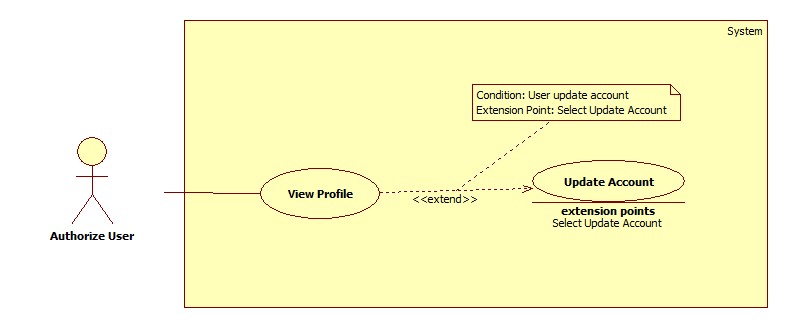
****

Figure : <Authorized User> View Profile

**Use case specification.**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM017** | |  |  |
| **Use Case No.** | ECRM017 | **Use Case Version** | 2.0 |
| **Use Case Name** | View Profile |  |  |
| **Author** | Tran Vinh Quang |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:** User | |  |  |
| **Summary:** This use case allows user to view all information of their account.  **Goal:** View information of account.  **Triggers:** User clicks on account name link.  **Preconditions:**   * User has successfully logged in system.   **Post Conditions:**   * **Success:** User will be see all of information of their account * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User view account’s information by send request to get profile. | System navigates to profile page.  It contains the following information:   * Username: Label * Họ Tên: Text input * “Số Điện Thoại”: Text input * “Trạng thái”: Label * “Cập nhật”: command * “Thay đổi Password”: command.   [Exception 1] |     **Alternative Scenario:** N/A  **Exceptions:**     |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Connection error | Show error message: “Lỗi kết nối, vui lòng tải lại” |   **Relationships:** extend Update Account  **Business Rules:**   * From Profile Page, User can change their password, update new phone number and update real name. * User can edit text “Số Điện Thoại” and send “Cập nhật” command to update new phone number. * User can edit text “Họ Tên” and send “Cập nhật” command to update new real name | | | |

Table : <Authorized User> View Profile

* + - 1. <Authorized User> Update Account

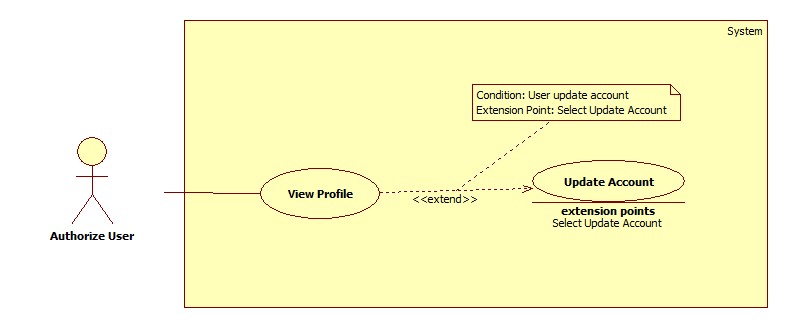
****

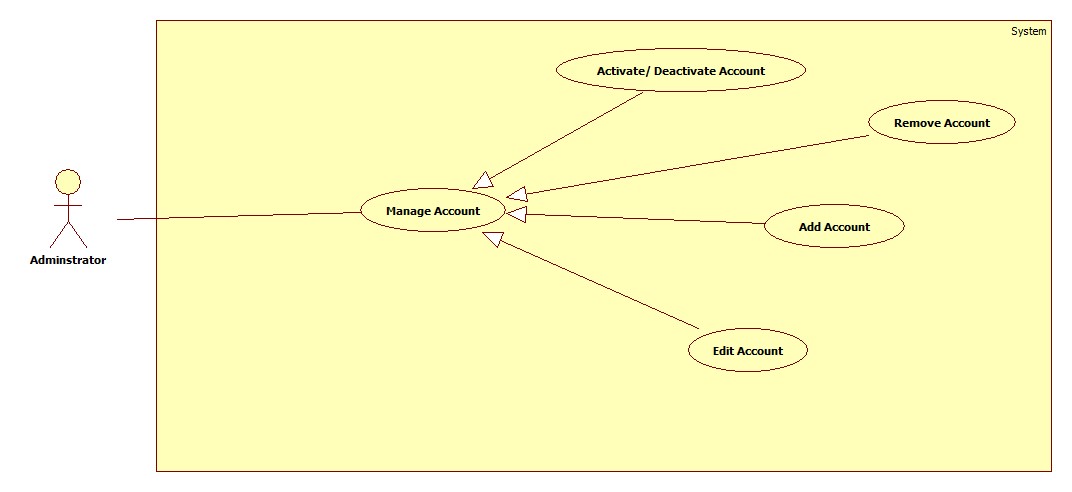
Figure : <Authorized User> Update Account

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM018** | |  |  |
| **Use Case No.** | ECRM018 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update Account |  |  |
| **Author** | Tran Vinh Quang |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:** User | |  |  |
| **Summary:** This use case allows user to change password of their account.  **Goal:** Change password of account.  **Triggers:** User send “Cập nhật Password” request in Profile page.  **Preconditions:**   * User has successfully logged in system.   **Post Conditions:**   * **Success:** User will be changed their password, show successful message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User send “Change Password” request in Profile page. | System navigates to “Change Password” page.  It contains form with the following information:   * Current Password: Text input, type: Password * New Password: Text input, min length: 6, max length: 20, type: Password, required. * Confirm Password: Text input, min length: 6, max length: 20, type: Password, required. * “Thay đổi”: command | | 2 | User fills information into the form and send “Thay đổi” request. | System show successful message and redirect to Profile page.  [Exception 1, 2, 3,4,5] |     **Alternative Scenario:** N/A  **Exceptions:**     |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Connection error | Show error message: “Lỗi kết nối, vui lòng tải lại” | | 2 | Current password field is wrong | Show error message: “Sai password hiện tại, vui lòng nhập lại” | | 3 | New password and confirm password are not match | Show error message: “New Password and Confirm password không giống nhau, vui lòng nhập lại” | | 4 | New password and confirm password field data length is longer than 20 character or lower than 6 character. | Show error message: “Password phải nằm trong khoản từ 6 đến 20 ký tự” | | 5 | New password is same with current password | Show error message: “Mật khẩu mới phải khác mật khẩu hiện tại” |   **Relationships:** N/A  **Business Rules:**   * If the first time user login to system, they will be redirect to “Change Password” page. * New password does not same with current password. | | | |

Table : <Authorized User> Update Account

* + - 1. **<Administrator> Use case Overview**

****

* + - 1. **< Administrator> Add Account**

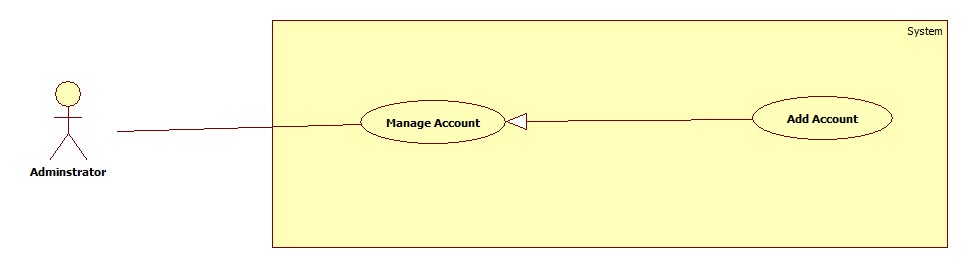
****

Figure : <Administrator> Add Account

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM019** | |  |  |
| **Use Case No.** | ECRM019 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add Account |  |  |
| **Author** | Tang Viet Hung |  |  |
| **Date** | 21/05/2015 | **Priority** | Normal |
| **Actor:** Staff | |  |  |
| **Summary:** This use case allows staff to add new account.  **Goal:** Add new account to the system.  **Triggers:** Staff wants to add new account.  **Preconditions:**   * User login as staff role.   **Post Conditions:**   * **Success:** New account will be created and show successful message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff focuses on “Account” and clicks “Tạo Tài Khoản” | System navigates to “Tạo Tài Khoản” page. Page includes form:   * Username: text input, min length: 6, max length: 30, required. * Password: text input, min length: 6, max length: 20, required * Họ tên: text input, min length: 6, max length: 50, required. * Số điện thoại: text input, min length: 10, max length: 11, type: number, required. * Chức vụ: list value: “User”, “Staff”, required. * Clear: command. * Tạo tài khoản: command   [Alternative 1] | | 2 | Staff enters information and then sends “Tạo tài khoản” request. | Create new account and show successful message: “Tài khoản được tạo thành công”.[Exception 1, 2, 3, 4, 5, 6] |     **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff sends Clear request. |  | |  | System will reset all data had inputted. |       **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Length of “username” is not in range. | Show unsuccessful message: “Username bao gồm từ 6 đến 10 ký tự. Xin thử lại!” | | 2 | “Số điện thoại” is empty or contain characters. | Show unsuccessful message: “Số điện thoại không hợp lệ. Xin thử lại!” | | 3 | “Họ tên” is empty. | Show unsuccessful message: “Họ tên không được bỏ trống. Xin thử lại!” | | 4 | “Username” have been already existed. | Show unsuccessful message: “Username đã tồn tại, vui lòng nhập lại!” | | 5 | “Số điện thoại” field data length is not in range | Show unsuccessful message: “Số điện thoại không hợp lệ. Xin thử lại!” | | 6 | “Password” field data length is not in range | Show unsuccessful message: “Password không hợp lệ. Xin thử lại!” |     **Relationships:** N/A  **Business Rules:**   * Account is created with active status. * “Số điện thoại” field data must start by “0”. * Staff only creates account with role user. * Admin can create account with role user or staff. | | | |

Table : <Administrator> Add New Account

* + - 1. <Administrator> Activate/Deactivate Account

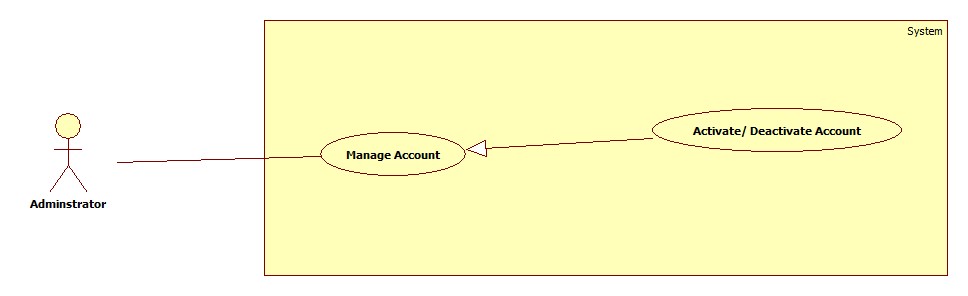


Figure : <Administrator> Activate/Deactivate Account

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM020** | |  |  |
| **Use Case No.** | ECRM020 | **Use Case Version** | 2.0 |
| **Use Case Name** | Activate/ Deactivate Account |  |  |
| **Author** | Tang Viet Hung |  |  |
| **Date** | 21/05/2015 | **Priority** | Normal |
| **Actor:** Staff | |  |  |
| **Summary:** This use case allows staff to activate and deactivate user’s account in system.  **Goal:** Activate and deactivate user’s account  **Triggers:** Staff wants to activate or deactivate user’s account.  **Preconditions:**   * User must login with staff role.   **Post Conditions:**   * **Success:** user’s account will be changed its status to Activated or Deactivated and show successful message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff focuses on “Account” and clicks “Edit Account” | System navigates to “Manage Account” page.   * Table contains columns:   + - Username.     - Classroom: teacher’s classroom.     - Check box. * Activate: command. * Deactivate: command. | | 2 | Staff selects account that they want to activate or deactivate. |  | | 3 | Staff sends “Activate” or “Deactivate” request. | Update status of account and show successful message: “Cập nhật tài khoản thành công!”  [Exception 1] |     **Alternative Scenario:** N/A    **Exceptions:**     |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff didn’t select any check box and then clicks “Activate” or “Deactivate” button | Show error message: “Vui lòng chọn ít nhất một tài khoản!” |   **Relationships:** N/A  **Business Rules:**   * After receiving request, system will change user’s status to “Activated” or “Deactivated”. * Only activated user can login to the system. | | | |

Table : <Administrator> Activate/Deactivate Account

* + - 1. <Administrator> Update Account

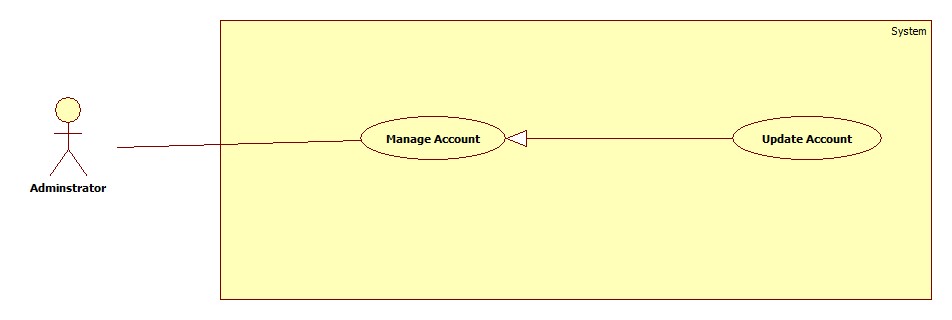


Figure : <Administrator> Update Account

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM021** | |  |  |
| **Use Case No.** | ECRM021 | **Use Case Version** | 2.0 |
| **Use Case Name** | Update Account |  |  |
| **Author** | Tran Vinh Quang |  |  |
| **Date** | 05/21/2015 | **Priority** | Normal |
| **Actor:** User | |  |  |
| **Summary:** This use case allows user to change information includes password, real-name, phone number, role of account.  **Goal:** Change information includes password, real-name, phone number, role of account.  **Triggers:** Staff clicks on “Account” link in Account manage page.  **Preconditions:**   * Staff has successfully logged in system.   **Post Conditions:**   * **Success:** Account will be changed information, show successful message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff clicks “Account” link. | System navigates to “Update Account” page.  It contains form with the following information:   * Username: textbox, min length: 6, max length: 30, required. * Password: textbox, min length: 6, max length: 20, required * Họ tên: textbox, min length: 6, max length: 50, required. * Số điện thoại: textbox, min length: 10, max length: 11, type: number, required. * Chức vụ: dropdown list, required. * Thay đổi: button | | 2 | User fills information into the form and sends “Thay đổi” requet. | System show successful message and redirect to Profile page.  [Exception 1, 2, 3, 4, 5, 6] |     **Alternative Scenario:** N/A  **Exceptions:**     |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Length of “username” is not in range. | Show unsuccessful message: “Username bao gồm từ 6 đến 10 ký tự. Xin thử lại!” | | 2 | “Số điện thoại” is empty or contain characters. | Show unsuccessful message: “Số điện thoại không hợp lệ. Xin thử lại!” | | 3 | “Họ tên” is empty. | Show unsuccessful message: “Họ tên không được bỏ trống. Xin thử lại!” | | 4 | “Username” have been already existed. | Show unsuccessful message: “Username đã tồn tại, vui lòng nhập lại!” | | 5 | “Số điện thoại” field data lenth is not in range | Show unsuccessful message: “Số điện thoại không hợp lệ. Xin thử lại!” | | 6 | “Password” field data ength is not in range | Show unsuccessful message: “Password không hợp lệ. Xin thử lại!” |   **Relationships:** N/A  **Business Rules:**   * Staff only update information of user * Admin can update information of staff and user. | | | |

Table : <Administrator> Update Account

* + - 1. **<Administrator> Remove Account**

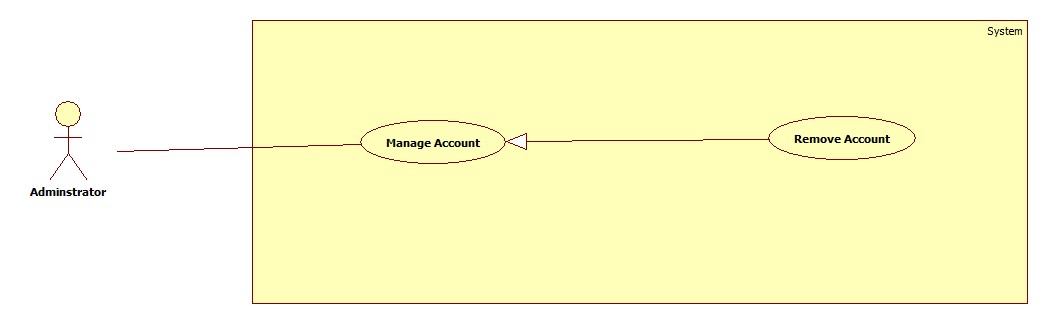
****

Figure : <Administrator> Remove Account

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – ECRM022** | |  |  |
| **Use Case No.** | ECRM022 | **Use Case Version** | 2.0 |
| **Use Case Name** | Remove Account |  |  |
| **Author** | Tang Viet Hung |  |  |
| **Date** | 21/05/2015 | **Priority** | Normal |
| **Actor:** Staff | |  |  |
| **Summary:** This use case allows administrator to remove user’s account in system.  **Goal:** Remove user’s account  **Triggers:** Administrator wants to remove user’s account.  **Preconditions:**   * User must login with administrator role.   **Post Conditions:**   * **Success:** user’s account will be remove from system and show successful message. * **Fail:** Unsuccessful message will be shown.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Administrator focuses on “Account” | System navigates to “Manage Account” page.   * Table contains columns:   + - Username.     - Classroom: teacher’s classroom.     - Check box. * Remove: command | | 2 | Administrator selects account that they want to remove |  | | 3 | Administrator sends “remove” request. | Update status of account and show successful message: “Xóa tài khoản thành công!”  [Exception 1] |     **Alternative Scenario:** N/A    **Exceptions:**     |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Staff didn’t select any check box and then send “Remove” command | Show error message: “Vui lòng chọn ít nhất một tài khoản!” |   **Relationships:** N/A  **Business Rules:**   * After receiving request, system will change user’s status to “removed” * Only activated user can login to the system. | | | |

Table : <Administrator> Remove Account

* + - 1. **<System> Use case Overview**

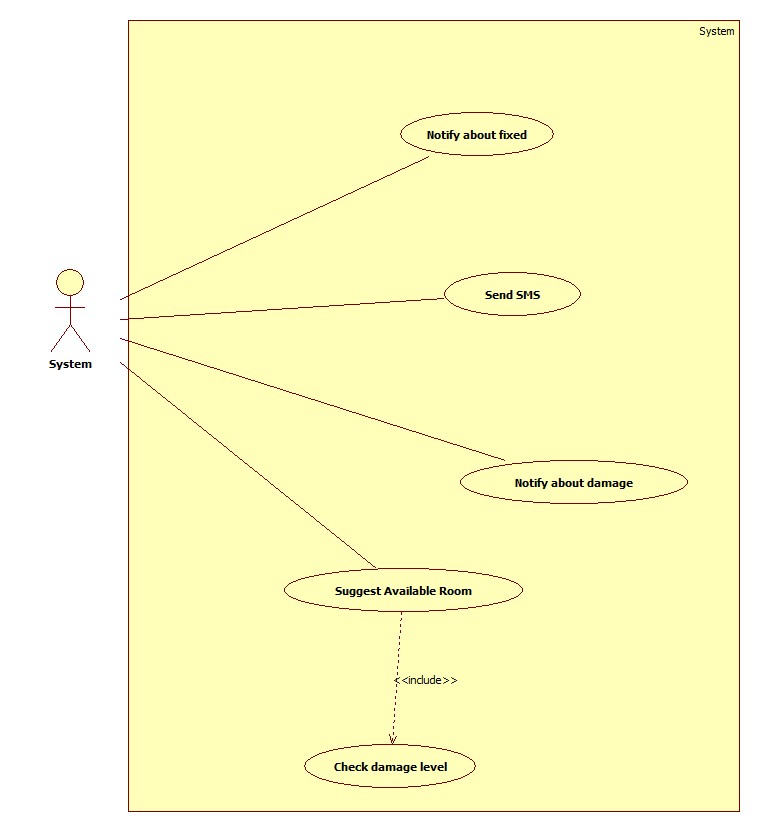
****

Figure : <System> Use case Overview

* + - 1. **<System> Notify about fixed**

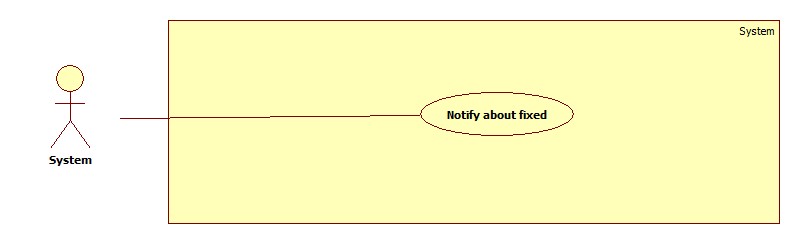
****

Figure : <System> Notify about fixed

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – ECRM023 | | | |
| Use Case No. | ECRM023 | **Use Case Version** | 2.0 |
| Use Case Name | Notify about fixed | | |
| Author | Tran Vinh Quang | | |
| Date | 21/05/2015 | **Priority** | Medium |
| Actor:   * System   Summary:   * This use case describes how the system notifies to users when necessary.   Goal:   * System will send notification to users.   Triggers:   * System run a timer task that trigger check event.   Preconditions:   * Their must be at least one report has resolved in the system.   Post Conditions:   * Success: Show the status of report and notify to user. * Fail: System display error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System run timer task to check report state and sends notify | System response:   * Notifications that need to be sent to users.   [Exception 1] |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | System timer task is interrupted | No notification will be sent. Error detail will be tracked in a log file. |   Relationships: N/A  Business rules:   * System will send notify after report has resolved. | | | |

Table : <System> Notify about fixed

* + - 1. **<System> Send SMS**

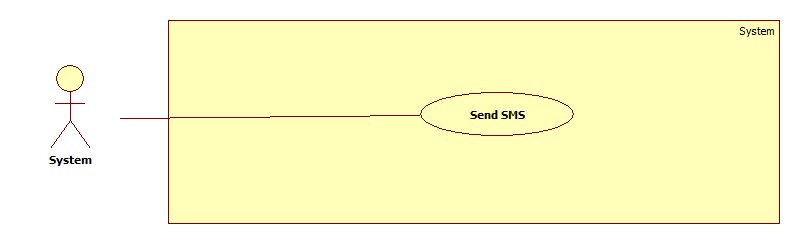
****

Figure : <System> Send SMS

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – ECRM024 | | | |
| Use Case No. | ECRM024 | **Use Case Version** | 2.0 |
| Use Case Name | Send SMS | | |
| Author | Tran Vinh Quang | | |
| Date | 21/05/2015 | **Priority** | Medium |
| Actor:   * System   Summary:   * This use case describes how the system send SMS to users when necessary.   Goal:   * System will send SMS to users.   Triggers:   * System run a timer task that trigger check event.   Preconditions:   * Their must be at least one report has resolved in the system.   Post Conditions:   * Success: Show the status of report and notify to user. * Fail: System display error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System run timer task to check report state and sends SMS | System response:   * SMS that need to be sent to users.   [Exception 1] |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | System timer task is interrupted | No notification will be sent. Error detail will be tracked in a log file. |   Relationships: N/A  Business rules:   * System will send SMS after report has resolved by changing room. * SMS will send to security and teacher. | | | |

Table : <System> Send SMS

* + - 1. **<System> Suggest Available Room**

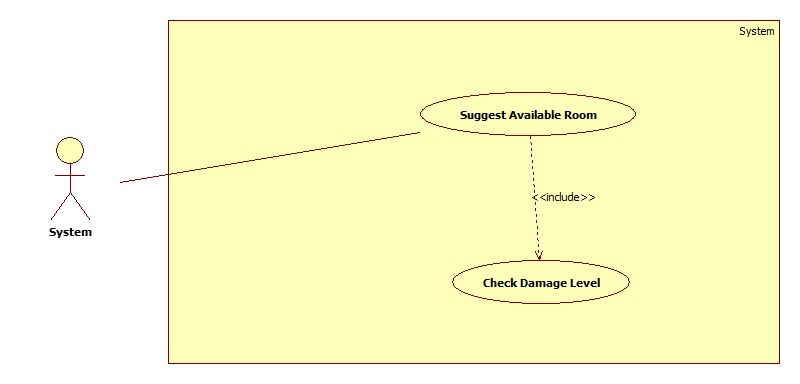
****

Figure : <System> Suggest Available Room

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – ECRM024 | | | |
| Use Case No. | ECRM024 | **Use Case Version** | 2.0 |
| Use Case Name | Sugges Available Room | | |
| Author | Tran Vinh Quang | | |
| Date | 21/05/2015 | **Priority** | Medium |
| Actor:   * System   Summary:   * This use case describes how the system suggests room to staff when necessary.   Goal:   * System will automatic check damage level and suggest room if necessary.   Triggers:   * System run a timer task that trigger check event.   Preconditions:   * Their must be at least one report has sent in the system.   Post Conditions:   * Success: Show the status of report and notify to user. * Fail: System display error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System run timer task to check damaged level. | System response:   * Damage level of report * System suggest available room if necessary   [Exception 1] | | 2 |  |  |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | System timer task is interrupted | No notification will be sent. Error detail will be tracked in a log file. |   Relationships: include Check damage level  Business rules:   * If damaged level is larger than 50%, system will automatic suggest available room * If damaged level is less than 10%, system will notify user after they send report. | | | |

Table : <System> Suggest Available Room

* + - 1. **<System> Notify about damaged**

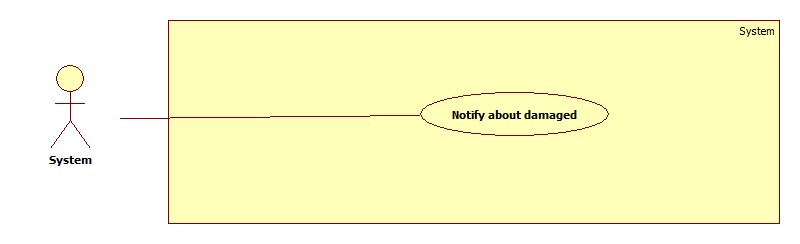
****

Figure : <System> Notify about damaged

**Use case specification**

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – ECRM025 | | | |
| Use Case No. | ECRM025 | **Use Case Version** | 2.0 |
| Use Case Name | Notify about damaged | | |
| Author | Tran Vinh Quang | | |
| Date | 21/05/2015 | **Priority** | Medium |
| Actor:   * System   Summary:   * This use case describes how the system notifies to staff when necessary.   Goal:   * System will send notification to staff.   Triggers:   * System run a timer task that trigger check event.   Preconditions:   * Their must be at least one report has sent in the system.   Post Conditions:   * Success: Show the status of report and notify to user. * Fail: System display error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System run timer task to check report state and sends notify | System response:   * Notifications that need to be sent to staff.   [Exception 1] |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | System timer task is interrupted | No notification will be sent. Error detail will be tracked in a log file. |   Relationships: N/A  Business rules:   * System will send notify after report has sent. | | | |

Table : <System> Notify about damaged

* 1. Software System Attribute
     1. **Usability**
        1. **Graphic User Interface**
* All the texts, labels should be written in Vietnamese
  + - 1. **Usability**
* Staff should need less 2 days of training to be productive with the system
* User need 2 hours of training to use mobile app or web application
* User Manual should be clear, easy to understand and increase experience for user.
  + - 1. **Installation**
* User can follow installation and manual guide for installation. If there are any problems, they can contact developer for help.
  + 1. **Reliability**
* System will notify staff about damaged equipment.
* System will notified or send SMS to user about fixing equipment.
* System will auto check about damage level to suggest how to resolve it.
* System will auto check status of room and change room if needed.
  + 1. **Availability**
* System should be online 24/7
  + 1. **Security**
* Privacy: Each role of user has a specific permission to interact with system
* System always checks authorization and authentication before doing anything
* All input data are validated before insert to database.
  + 1. **Maintainability**
* The system is divided into separated modules
* The code is easy to maintain and upgrade
  + 1. **Portability**
* Admin, Staff, User can use web application on every OS supported web browser.
* For mobile application, they must use Android smartphones that have version greater than 4.0.
  + 1. **Performance**
* The speed of suggestion should be less than 10 seconds
* The speed of report from user to staff should be less than 10 seconds
  1. **Conceptual Diagram**

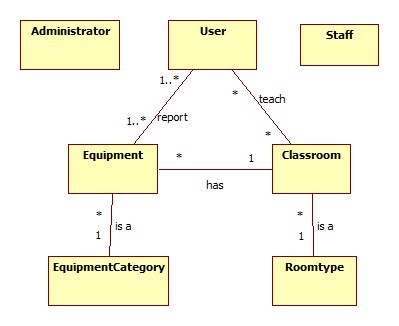


Figure : Conceptual Diagram

1. **Software Design Description**
   1. **Design Overview**

* This document describes the technical and user interface design of ECRM system. It includes the architectural design, the detailed design of common functions, the business functions and the design of database model.
* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
  + Section 1: gives an overall overview of this document.
  + Section 2: gives an overall description of the system architecture design.
  + Section 3: gives component diagrams that describe the connection and integration of the system.
  + Section 4: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
  + Section 5: describe a fully attributed Entity Relationship Diagram
  1. **System Architecture Design**

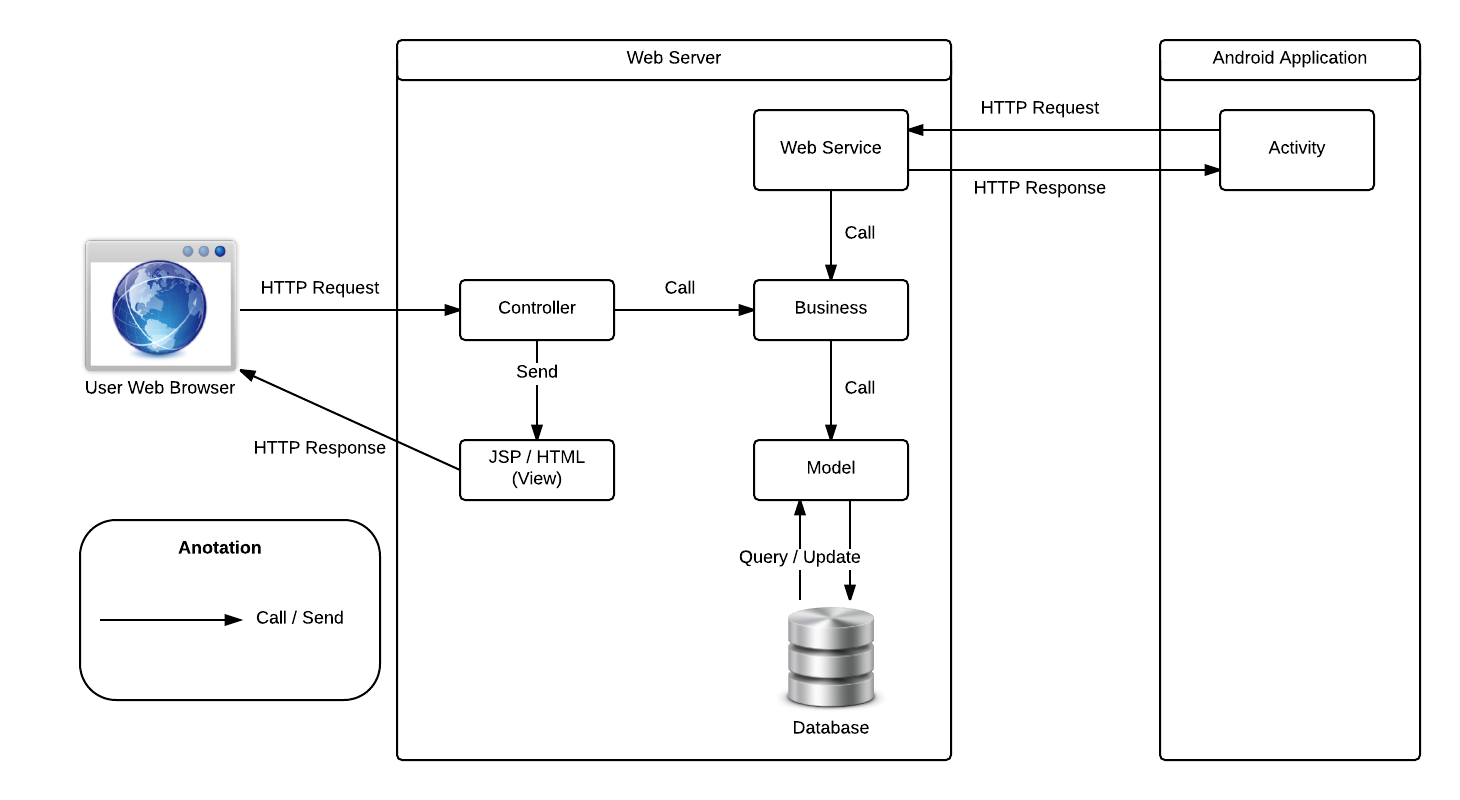
****

Figure : System Architecture Design

* + 1. **Web Application architecture description**

In Web Application, the system is developed under Spring MVC architecture style. We choose this architecture for Web application because of following advantages:

* Web app contains a Web service (public API for mobile app) with MVC architecture we can separate business code with Controller and View so we can use the business code in web service without repeat the code.
* In scope of 3-member team this, MVC architecture makes it easier to split the big project into small modules and make it easier to assign each module for members in our team.

This project follows MVC architecture with following components:

* **Web Service:** is the part of the application that acts like event handler for web and mobile communication via REST method.
* **Controller:** is the part of the application that acts like event handler to handles user interaction. Typically controller read data from a request and calls appropriate Business’s method then selects view to return to user.
* **JSP/HTML (View):** is the part of the application that handles the display of the data. The selection of View is under control of Controller.
* **Business:** is the part of the application that does business processing to solve domain problems.
* **Model:** is the part of the application that acts like a data transfer object between the system and database.
  + 1. **Mobile Application architecture description**

The application is developed as an Android native application. In general, the application architecture conforms to Android architecture.



Figure : Android Application Architecture

**Reference:** [Android Developer Guide - Application Fundamentals](http://developer.android.com/guide/components/fundamentals.html)

This project follows Android application architecture with following components:

* **Activity** is the basic core of an android application that handles user input, create thread to run asynchronous tasks, send request and receive data from server via web services ...
  1. **Component Diagram**

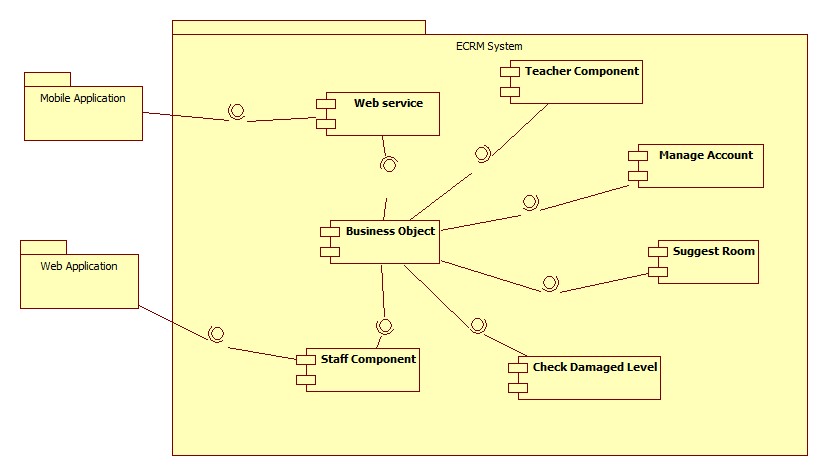


Figure : Component Diagram

|  |  |
| --- | --- |
| Component Dictionary: Describes components | |
| Web Application | Web application package: View, Controller. |
| Mobile Application | Mobile application package. |
| Web Service | Include all web API controllers of the system. Include all web API controllers of the system |
| Staff Component | Component to handle staff activities in the system |
| Teacher Component | Component to handle teacher activities in the system |
| Manage Account | Business logic for manage account processing. |
| Business Objects | Handle business operations for every component. |
| Suggest Room | Business logic for suggest room processing. |
| Check Damaged Level | Business logic for checking damaged level processing. |

Table : Component Dictionary

* 1. **Detail Description**
     1. **Class Diagram**
     2. **Class Diagram Explanation**
     3. **Interactive Diagram**
        1. **Web Application**
           1. **Staff**
           2. **Teacher**
           3. **Guest**
           4. **Admin**
        2. **Mobile Application**
           1. **ECRM Staff**
           2. **ECRM Teacher**
  2. **Interface**
     1. **Component Interface**
        1. **Web Service Interface**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Signature | Description | Input | Output | Output Format | Exception |
| public ResultDTO checkConnection() | Check server status | N/A | Json Boolean the status of server | Boolean | JsonProcessingException |
| Contractlean if the update is successntractpublic AccountDTO Login(String username, String password) | Login to system with account r | Request contains:  Username: String  Password: String | Json AccountEntity object | AccountEntity | JsonProcessingException  NoResultException |
| public List<CategoryDTO> getCategory(String username) | Get all equipment category in system | Request contains:  Username: String | Json String List of CategoryDTO | List CategoryEntity | JsonProcessingException  NoResultException |
| public List<ScheduleDTO> getSchedule(String username) | Get schedule in day of account | Request contains:  Username: String | Json String List of ScheduleDTO | List ScheduleEntity | JsonProcessingException  NoResultException |
| public List<EquipmentClassDTO> getEquipment(Integer classId) | Get equipment of classroom | Request contains:  classId: Integer | Json String List of EquipmentClassDTO | List EquipmentEntity | JsonProcessingException  NoResultException |
| public ResultDTO createReport(Report r) | Create report from mobile | Request object r contains:  Username: String  classId: Integer  listDamaged: String  listPosition: String  listDescription: String  evaluate: String  listEvaluate: String  createTime: String | Json String ResultDTO | ResultEntity | JsonProcessingException  NoResultException |
| public List<ReportDTO> getReportByUsername(String username, Integer offset, Integer limit) | Get limit report of user has username from offset | Request contains:  Username: String  Offset: Integer  Limit: Integer | Json String List ReportDTO | List ReportEntity | JsonProcessingException  NoResultException |
| public List<ReportClassDTO> getReportStaff(String status, Integer limit, Integer offset) | Get limi report of staff with status from offset | Request contains:  Status: String  Limit: Integer  Offset: Integer | Json String List ReportDTO | List ReportClassEntity | JsonProcessingException  NoResultException |
| public List<String> getAvailableRoom(Integer classId) | Get available room with same options with current room. | Request contains:  classId: Integer | Json String List Classroom | List String Classroom | JsonProcessingException  NoResultException |
| public ResultDTO changeRoom(IInteger from, Integer to) | Change schedule of user from current room to new room | Request contains:  from: Integer  to: Integer | Json String ResultDTO | ResultEntity | JsonProcessingException. |
| public ClassDTO getClassroom(Integer classId) | Get properties of classroom to save local | Request contains:  classId: Integer | Json String ClassDTO | ClassEntity | JsonProcessingException  NoResultException |
| public List<ScheduleDTO> getAllSchedule(String username) | Get all schedule in week of user | Request contains:  Username: String | Json String List ScheduleDTO | ScheduleEntity | JsonProcessingException  NoResultException |
| public ResultDTO remove(String username, Integer reportId) | Remove report | Request contains:  Username: String  reportId: Integer | Json String ResultDTO | ResultEntity | JsonProcessingException  NoResultException |
| public ResultDTO editReport | Edit report | Request contains:  reportId: String  Username: String  classId: Integer  listDamaged: String  listPosition: String  listDescription: String  evaluate: String  listEvaluate: String  createTime: String | Json String ResultDTO | ResultEntity | JsonProcessingException  NoResultException |
| public ResultDTO sendNotification (String message, String ListUser) | Send message to listUser by GCM notification | Request contains:  Message: String  ListUser: String | Json String ResultDTO | ResultEntity | JsonProcessingException  NoResultException |
| public ResultDTO sendSMS (String message, String listUser) | Send message to listUser by SMS | Request contains:  Message: String  ListUser: String | Json String ResultDTO | ResultEntity | JsonProcessingException  NoResultException |
| public boolean checkSchedule(Integer classId) | Check class has schedule after current time or not | Request contains:  classId: Integer | Json Boolean | Boolean | JsonProcessingException  NoResultException |
| public ResultDTO resolveReport (String listRoom) | Resolve report by room | Request contains:  listRoom: String | Json String ResultDTO | ResultEntity | JsonProcessingException  NoResultException |

Table : Web Service Interface

**Entity**

|  |  |
| --- | --- |
| Entity | Description |
| ResultEntity | Integer error\_code  String error |
| AccountEntity | String username  String password  String fullname  String phone  String role  String lastLogin  String status |
| CategoryEntity | String name  String imageUrl |
| ScheduleEntity | Integer classId  String className  String timeFrom  String timeTo  String date |
| ReportEntity | Integer reportId  String username  String fullname  Integer classId  String className  String createTime  String evaluate  Boolean status  Integer damageLevel  String changedRoom  List<ReportDetailEntity> listEquipments |
| EquipmentEntity | String equipmentName  String timeRemain  String company  Boolean isDamaged |
| ReportDetailEntity | String equipmentName  String description  String damaged  Boolean status  String solution  Timestamp resolveTime |
| ReportClassEntity | Integer roomId  String roomName  String timeReport  Integer damageLevel  String evaluate  String userReport  Integer systemEvaluate  String changeRoom  List<EquipmentReportEntity> listEquipments |
| EquipmentReportEntity | Integer reportId  Integer equipmentId  String equipmentName  Integer quantity  Boolean status  String evaluate  String damage |
| ClassEntity | Integer classId  String className  Integer damageLevel |

Table : Web Service Interface - Entity

**Exceptions**

|  |  |
| --- | --- |
| Exception | Description |
| JsonProcessingException | encountered when processing (parsing, generating) JSON content that are not pure I/O problems |
| NoResultException | Thrown by the persistence provider when getSingleResult() is executed on a query and there is no result to return |

Table : Web Service Interface - Exceptions

* + 1. **Web Application Design**
       1. **User Interface Design**
       2. **Staff Interface Design**
       3. **Admin Interface Design**
    2. **ECRM Staff Design**
       1. **Login Screen**
       2. **List Report Newest Screen**
       3. **List Report On Going Screen**
       4. **List Report Finished Screen**
       5. **Report Detail Screen**
       6. **Change Room Screen**
       7. **Resolve Report Screen**
    3. **ECRM Teacher Design**
       1. **Login Screen.**

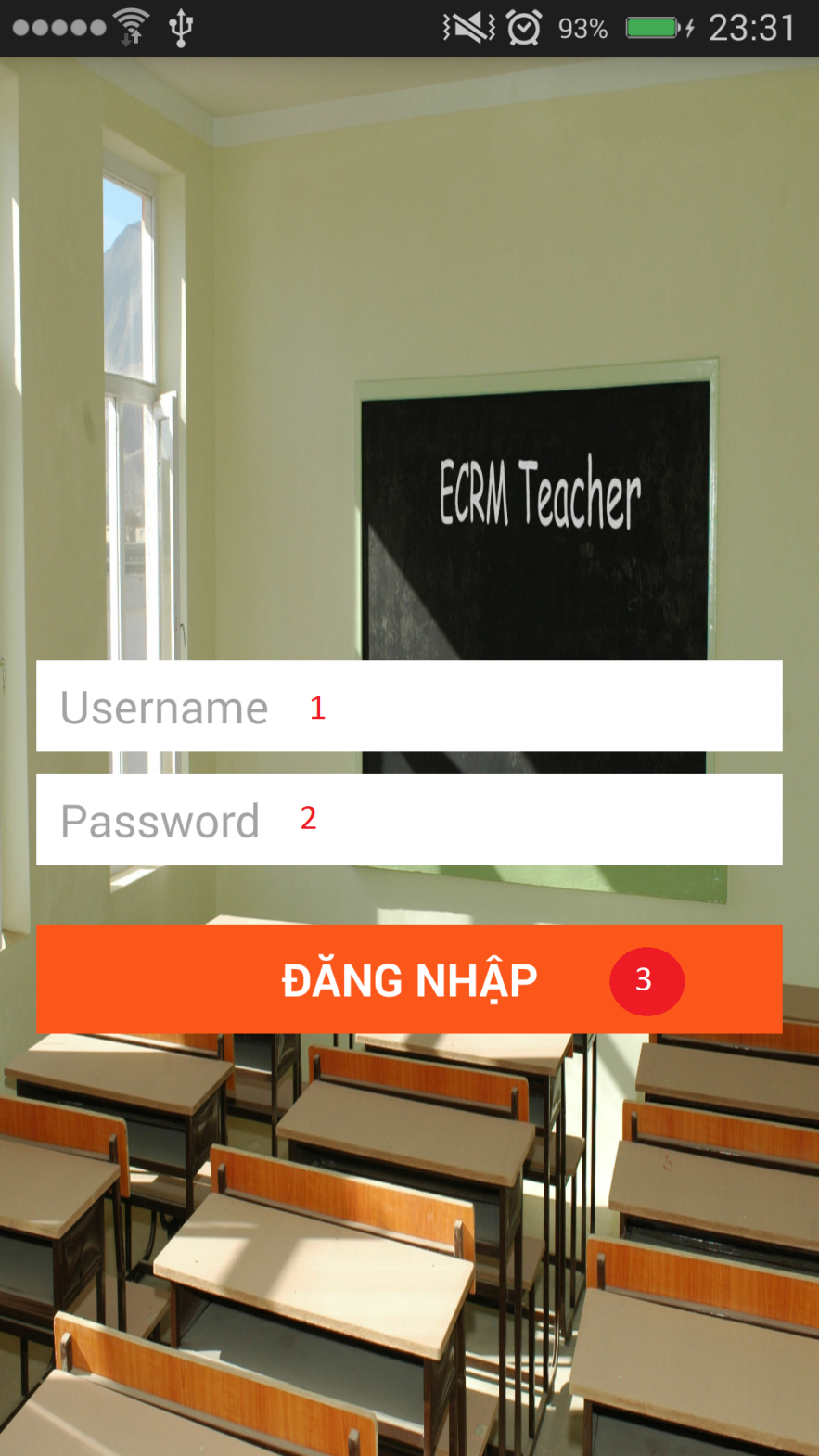
****

Figure : ECRM Teacher Login Screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | txtUsername | Username of account to login to application | No | No | Textbox | String | 0 - 35 |
| 2 | txtPassword | Password of account to login to application | No | No | Textbox | String | 0 - 35 |

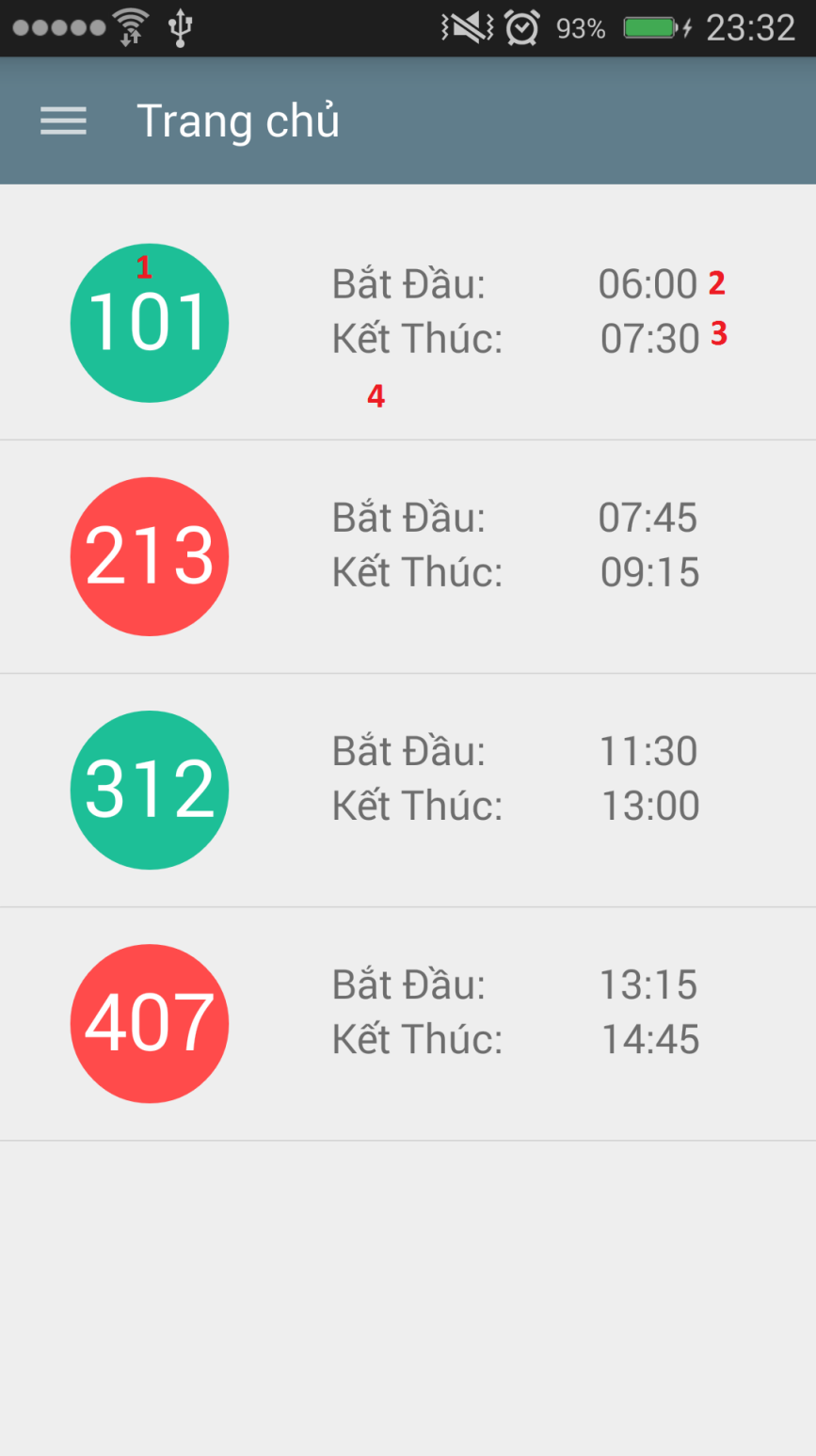
Table : Login Screen – Fields

**Buttons**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Function | Description | Validation | Outcome |
| 3 | btnLogin | Login to application with username and password in textbox | No | Show list schedule of this account |

Table : Login Screen - Buttons

* + - 1. **Schedule Screen**

****

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Description | Read only | Mandatory | Control Type | Data Type | Length |
| 1 | lblRoomName | Name of classroom | Yes | Yes | Label | String | 0 - 35 |
| 2 | lblTimeFrom | Time start of schedule | Yes | Yes | Label | String | 0 - 35 |
| 3 | lblTimeTo | Time end of schedule | Yes | Yes | Label | String | 0 - 35 |

Table : Schedule Screen - Fields

**Buttons**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Function | Description | Validation | Outcome |
| 4 | lstSchedule | Click to a row from list to create report in this classroom | N/A | Show list equipment of this classroom |

Table : Schedule Screen - Buttons

* + - 1. **View Report History Screen**
      2. **Create Report Screen**
      3. **Change Password Screen**
      4. **Report Detail Screen**
  1. **Database Design**
     1. **Entity Relationship Diagram**
     2. **Entity Dictionary**
  2. **Algorithms**

1. **System Implementation & Test**
   1. **Introduction**
      1. **Overview**

This describes the approach and methodologies used by group to plan, organize and manage the testing of ECRM system. It provides in the detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

* + 1. **Test Approach**
* Goal: Test all features in the whole ECRM system based on the core flow.
* Method: black-box testing
* Technique: check list

The testing for this project will consists of Integration System test level. Testing the program which was integrated and as a complete system to ensure that the software requirements have been met.

* Integration testing would be performed by all member of team and approved by team leader.
* System testing is focused on assessing the system’s reliability. This process is concerned with finding errors that result from unanticipated interactions between components and component interface problems.
  1. **Database Relationship Diagram**
     1. **Physical Diagram**

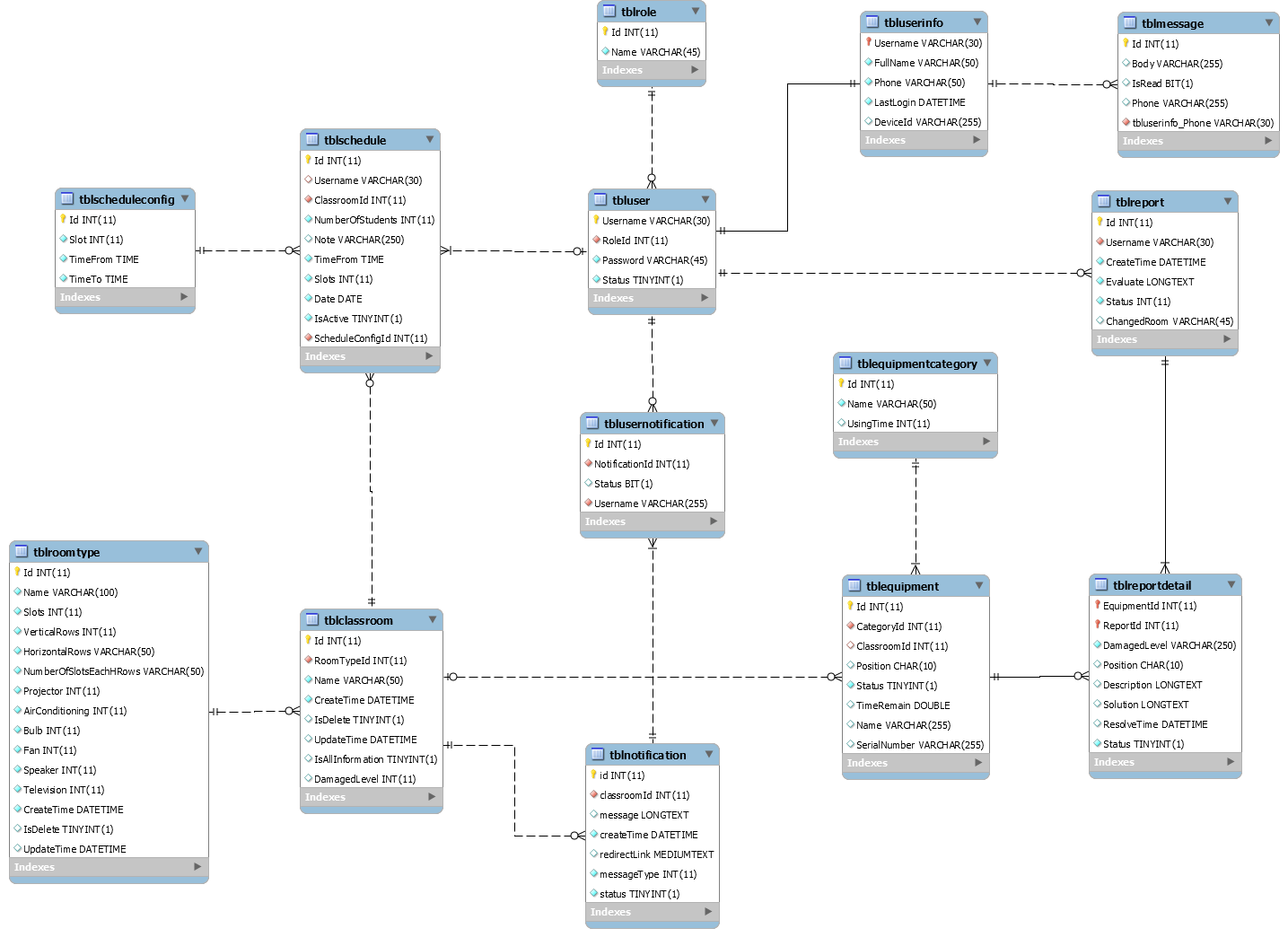


Figure : Physical Diagram

* + 1. **Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Attributes | Description | Domain | Null |
| tblUser | Username | Username to login system | VARCHAR(30) | No |
|  | Password | Password of account to login system | VARCHAR(45) | No |
|  | Status | Status of account | TINYINT(1) | No |
|  | RoleId | Role of account | INT(11) | No |
| tblRole | Id | Id of role | INT(11) | No |
|  | Name | Name of role | VARCHAR(45) | No |
| tblUserInfo | Username | Username of account | VARCHAR(30) | No |
|  | FullName | Fullname of account | VARCHAR(50) | No |
|  | Phone | Phone number of account | VARCHAR(50) | No |
|  | LastLogin | Last login of account to system | DATETIME | Yes |
|  | DeviceId | Device ID of account when using mobile application | VARCHAR(255) | Yes |
| tblMessage | Id | Id of message | INT(11) | No |
|  | Body | Content of message | VARCHAR(255) | No |
|  | IsRead | Status of message | BIT(1) | No |
|  | Phone | Phone number send message | VARCHAR (30) | No |
| tblReport | Id | Id of report | INT(11) | No |
|  | Username | Username of account create report | VARCHAR(30) | No |
|  | CreateTime | Time report has created | DATETIME | No |
|  | Evaluate | Evaluate of teacher about this report | LONGTEXT | No |
|  | Status | Status of report | INT(11) | No |
|  | ChangeRoom | Room number has changed | VARCHAR(45) | Yes |
| tblEquipmentCategory | Id | Id of equipment category | INT(11) | No |
|  | Name | Name of equipment category | VARCHAR(50) | No |
|  | UsingTime | Time using of equipment category | INT(11) | Yes |
| tblEquipment | Id | Id of equipment | INT(11) | No |
|  | CategoryId | Id of equipment category | INT(11) | No |
|  | ClassroomId | Id of classroom has equipment | INT(11) | No |
|  | Position | Position of equipment in classroom | CHAR(10) | No |
|  | Status | Status of equipment | TINYINT(1) | No |
|  | TimeRemain | Time remain of equipment | DOUBLE | No |
|  | Name | Name of equipment | VARCHAR(255) | No |
|  | SerialNumber | Serial Number of equipment | VARCHAR(255) | No |
| tblReportDetail | EquipmentId | Id of equipment has reported | INT(11) | No |
|  | ReportId | Id of report | INT(11) | No |
|  | DamagedLevel | Damage level of equipment | VARCHAR(250) | No |
|  | Position | Position of equipment in classroom | CHAR(10) | No |
|  | Description | Description of equipment | LONGTEXT | Yes |
|  | Solution | Solution to resolve equipment | LONGTEXT | Yes |
|  | ResolveTime | Time resolve this equipment | DATETIME | Yes |
|  | Status | Status of this report detail | TINYINT(1) | No |
| tblUserNotification | Id | Id of user notification | INT(11) | No |
|  | NotificationId | Id of notification | INT(11) | No |
|  | Status | Status of notification | BIT(1) | No |
|  | Username | Username of account received this notification | VARCHAR(30) | No |
| tblNotification | Id | Id of notification | INT(11) | No |
|  | ClassroomId | Id of classroom | INT(11) | No |
|  | Message | Content of notification | LONGTEXT | No |
|  | CreateTime | Time notification has been created | DATETIME | No |
|  | RedirectLink | Link will redirect after user click to notification | MEDIUMTEXT | No |
|  | MessageType | Type of message | INT(11) | No |
|  | Status | Status of notification | TINYINT(1) | No |
| tblClassroom | Id | Id of classroom | INT(11) | No |
|  | RoomTypeId | Id of roomtype | INT(11) | No |
|  | Name | Name of classroom | VARCHAR(50) | No |
|  | CreateTime | Time create of classroom | DATETIME | No |
|  | IsDelete | Check classroom has been deleted | TINYINT(1) | No |
|  | UpdateTime | Time update classroom | DATETIME | No |
|  | IsAllInformation | Check classroom has full equipment | TINYINT(1) | No |
|  | DamageLevel | Damage level of classroom | INT(11) | No |
| tblSchedule | Id | Id of schedule | INT(11) | No |
|  | Username | User of schedule | VARCHAR(30) | No |
|  | ClassroomId | Classroom of schedule | INT(11) | No |
|  | NumberOfStudents | Number of student of classroom | INT(11) | No |
|  | Note | Note of schedule | VARCHAR(250) | Yes |
|  | TimeFrom | Time start schedule | TIME | No |
|  | Slots | Total slot of schedule | INT(11) | No |
|  | Date | Date of schedule | DATE | No |
|  | IsActive | Status of schedule | TINYINT(1) | No |
|  | ScheduleConfigId | Config of schedule | INT(11) | No |
| tblScheduleConfig | Id | Id of schedule config | INT(11) | No |
|  | Slot | Slot of schedule config | INT(11) | No |
|  | TimeFrom | Time start of schedule config | TIME | No |
|  | TimeTo | Time end of schedule config | TIME | No |
| tblRoomType | Id | Id of roomtype | INT(11) | No |
|  | Name | Name of roomtype | VARCHAR(100) | No |
|  | Slots | Number of slot of roomtype | INT(11) | No |
|  | VerticalRows |  | INT(11) | No |
|  | HorizontalRows |  | VARCHAR(50) | No |
|  | NoSlotEachHRows |  | VARCHAR(50) | No |
|  | Projector | Check roomtype has projector or not | INT(11) | No |
|  | AirConditioning | Check roomtype has air condition or not | INT(11) | No |
|  | Bulb | Check roomtype has bulb or not | INT(11) | No |
|  | Fan | Check roomtype has fan or no | INT(11) | No |
|  | Speaker | Check roomtype has speaker or not | INT(11) | No |
|  | Television | Check roomtype has television or not | INT(11) | No |
|  | CreateTime | Time create roomtype | DATETIME | No |
|  | IsDelete | Status of roomtype is deleted or not | TINYINT(1) | No |
|  | UpdateTime | Update Time of roomtype | DATETIME | No |

Table : Physical Diagram - Data Dictionary

* 1. **Performance Measures**
     1. **Web Application Page Load Speed**
        1. **Definition**

This section tests the general page load speed from all the page of the web application

* + - 1. **Test environment**

**Server:**

* Operating System: Ubuntu 14.04 Server 32 bit
* RAM: 512MB
* Storage: 40GB
* Network: 1Mbps
  + - 1. **Test cases**

Using browser from client side and request each page of web application. Record the page load time using Chrome Dev Tools at client side.

Test pages:

|  |  |
| --- | --- |
| Public | Login Page |
| User | Dashboard |
| Create Report |
| View History Report |
| Staff | Dashboard |
| Schedule Config   * Add new schedule config * Update schedule config * Remove schedule config |
| Schedule   * Search schedule by user in default time. * Search schedule by classroom in default time. * Search between time with all user and classroom * Search between time will specific user and classroom * Search between time with specific user in all classroom. * Search between time with specific classroom with all user. * Import schedule excel file in 1 week of school (about 15KB) * Import manually 1 schedule |
| Statistic |
| Classroom   * Create new classroom * Remove classroom * Edit classroom * List all classroom * Search classroom by class name |
| Roomtype   * Create new roomtype * Remove roomtype * Edit roomtype * List all roomtype * Search roomtype by name |
| Equipment   * List all equipment * Create new equipment * Remove equipment |
|  | Report   * List all report * View report * Resolve report * Remove report |
| Admin | Account   * List all account * Activate/ Deactivate account * Create account * Edit account |

Table : Web Application Page Load - Test Pages

* + - 1. **Test result**

The test is run 10 times, in each time we go through all the pages listed above and log down the average page load time.

Figure : Web Application Page Load - Averange Page Load

|  |  |  |
| --- | --- | --- |
| Test No. | Average page load time (second) | Execute date |
| 1 | 1.85 | 25 July 2015 |
| 2 | 1.68 | 25 July 2015 |
| 3 | 2.94 | 25 July 2015 |
| 4 | 3.19 | 25 July 2015 |
| 5 | 2.63 | 25 July 2015 |
| 6 | 2.96 | 25 July 2015 |
| 7 | 2.48 | 25 July 2015 |
| 8 | 3.39 | 25 July 2015 |
| 9 | 2.73 | 25 July 2015 |
| 10 | 2.10 | 25 July 2015 |
|  | **Average: 2.59 (seconds)** |  |

Table : Web Application Page Load - Averange Page Load

* + 1. **Mobile Application API Load Speed**
       1. **Definition**

This section tests the load speed of mobile apps when connection to server through API

* + - 1. **Test environment**

**Server:**

* Operating System: Ubuntu 14.04 Server 32 bit
* RAM: 512MB
* Storage: 40GB
* Network: 1Mbps

**Client:**

|  |  |  |
| --- | --- | --- |
| Hardware Requirements | Minimum | Recommended |
| Internet Connection | 512Kbps | Wi-Fi Connection 12MB |
| Operating System | Android 4.0 | Android 4.0 |
| Memory | 128MB of RAM | 1GB of RAM or more |

* + - 1. **Test case**

Using mobile application to send the entire API request to web server.

List of APIs:

|  |  |  |
| --- | --- | --- |
| No. | API | Description |
| 1 | checkConnection | Check connection to server |
| 2 | login | Login to application |
| 3 | schedule | Get schedule of user |
| 4 | getEquipments | Get equipment of each classroom |
| 5 | createReport | Create new report |
| 6 | getCurrentTime | Get current time of server |
| 7 | getReportByUsername | Get report of user |
| 8 | remove | Remove report |
| 9 | editReport | Edit report |
| 10 | getReportStaff | Get all report of staff |
| 11 | getAvailableRoom | Get available room for change room function |
| 12 | changeRoom | Do change room function when needed |
| 13 | sendNotification | Send notification to user |
| 14 | sendSMS | Send SMS to user |
| 15 | map | Get map of classroom |
| 16 | getAllReport | Get all report |
| 17 | resolve | Resolve report |
| 18 | checkSchedule | Check classroom has user teacher after that slot. |
| 19 | getAllSchedule | Get all schedule of user in week |
| 20 | getClassroom | Get current status of classroom |

* + - 1. **Test result**

The test is run 10 times, each time we run all the test cases listed above and log down the average API response time.

|  |  |  |
| --- | --- | --- |
| Test No. | Average page load time (second) | Execute date |
| 1 | 3.04 | 25 July 2015 |
| 2 | 4.86 | 25 July 2015 |
| 3 | 3.78 | 25 July 2015 |
| 4 | 4.42 | 25 July 2015 |
| 5 | 4.45 | 25 July 2015 |
| 6 | 3.43 | 25 July 2015 |
| 7 | 4.05 | 25 July 2015 |
| 8 | 2.96 | 25 July 2015 |
| 9 | 4.13 | 25 July 2015 |
| 10 | 4.68 | 25 July 2015 |
|  | **Average: 3.98 (seconds)** |  |

* 1. **Test Plan**

The overall purpose of testing is to ensure ECRM system meets its entire technical, functional and business requirement. The purpose of this document is to describe the overall test plan and strategy for testing the ECRM system. The approach described in this document provides the framework for all testing related to this application. Individual test cases will be written for each version of the application that is released. This document will also be updated as required for each release.

* + 1. **Feature to be tested**
* Staff: View schedule, resolve report, remove report, change room, import file schedule, import schedule manual.
* User: Create report, edit report and remove report.
* Mobile Application: Create report, edit report, remove report, resolver report and change room.
* System: Automatically check damage and change room.
  + 1. **Feature no to be tested**
* Admin: Create account, update account, active/deactivate account.
* Staff: Create classroom, edit classroom, remove classroom, create roomtype, create equipment, edit equipment, edit roomtype, and delete roomtype, config schedule, statistic, search schedule.
* Login, logout, change profile
* View information: list roomtype, list classroom, list report.
  1. **System Testing Test Case**