SMART DOOR

SOFTWARE SYSTEM PROJECT REPORT

Phase 1

PREPARED BY:
RAMTIN RASSOLI, SALAR ARBABI

SHARIF UNIVERSITY OF TECHNOLOGY
TEHRAN, IRAN

DECEMBER 2015
1 Table of Contents

2 Table of figures .................................................................................................................. 5

3 List of tables....................................................................................................................... 6

4 Project Description............................................................................................................. 7
  4.1 Project Overview .......................................................................................................... 7
  4.2 The purpose of the project .......................................................................................... 7
    4.2.1 The user business or Background of the project .................................................. 7
    4.2.2 The context of the work ....................................................................................... 7
    4.2.3 Goals of the project ............................................................................................ 9
    4.2.4 Measurement ........................................................................................................ 9
  4.3 The scope of the work ................................................................................................. 10
    4.3.1 The current situation .......................................................................................... 10
    4.3.2 Competing products .......................................................................................... 14
  4.4 Product scenarios ......................................................................................................... 16
    4.4.1 Product scenario list .......................................................................................... 17
    4.4.2 Individual product scenarios ............................................................................. 17
  4.5 Stakeholders ................................................................................................................ 19
    4.5.1 The student ...................................................................................................... 19
    4.5.2 The professor ..................................................................................................... 19
    4.5.3 User participation ............................................................................................... 20
  4.6 Mandated constraints .................................................................................................. 21
    4.6.1 Solution constraints ......................................................................................... 21
    4.6.2 Implementation environment of the current system ......................................... 22
    4.6.3 Partner or collaborative applications ............................................................... 22
    4.6.4 Anticipated workplace environment ............................................................... 23
  4.7 Naming conventions and definitions .......................................................................... 23
    4.7.1 Definitions of key terms ................................................................................... 23
    4.7.2 Uml and other notation used in this document ............................................... 24
  4.8 Relevant facts and assumptions ................................................................................. 24
    4.8.1 Facts ................................................................................................................ 25
    4.8.2 Assumptions ..................................................................................................... 25

5 Requirements ..................................................................................................................... 25
  5.1 Product use cases ....................................................................................................... 25
    5.1.1 Use case diagrams ........................................................................................... 25
  5.2 Functional requirements ............................................................................................. 27
  5.3 Data requirements ....................................................................................................... 27
  5.4 Performance requirements .......................................................................................... 31
    5.4.1 Speed and latency requirements ....................................................................... 31
    5.4.2 Precision or accuracy ....................................................................................... 31
    5.4.3 Capacity requirements ....................................................................................... 31
  5.5 Dependability requirements ....................................................................................... 31
    5.5.1 Reliability requirements .................................................................................... 31
    5.5.2 Availability requirements .................................................................................. 31
5.5.3 Robustness or fault-tolerance requirements..............................................................31
5.5.4 Safety-critical requirements......................................................................................31
5.6 Maintainability and supportability requirements.....................................................32
  5.6.1 Maintenance requirements ......................................................................................32
  5.6.2 Supportability requirements ..................................................................................32
  5.6.3 Adaptability requirements .....................................................................................32
  5.6.4 Scalability requirements .......................................................................................32
  5.6.5 Longevity requirements .......................................................................................32
5.7 Security requirements ..................................................................................................33
  5.7.1 Access requirements ............................................................................................33
  5.7.2 Integrity requirements ..........................................................................................33
  5.7.3 Privacy requirements ............................................................................................33
5.8 Usability and humanity requirements........................................................................33
  5.8.1 Ease of use requirements .......................................................................................33
  5.8.2 Personalization and internationalization requirements .........................................33
  5.8.3 Learning requirements ..........................................................................................33
  5.8.4 Understandability and Politeness Requirements ..................................................34
  5.8.5 User documentation requirements ........................................................................34
  5.8.6 Training Requirements .......................................................................................34
5.9 Look and feel requirements.........................................................................................34
  5.9.1 Appearance requirements .....................................................................................34
  5.9.2 Style requirements ...............................................................................................34
5.10 Operational and Environmental Requirements .....................................................35
  5.10.1 Expected Physical Environment ..........................................................................35
  5.10.2 Productization Requirements ............................................................................35
  5.10.3 Release Requirements .......................................................................................35
5.11 Legal requirements ....................................................................................................36
6 Design ..........................................................................................................................37
  6.1 System design ............................................................................................................37
    6.1.1 Design goals ........................................................................................................37
  6.2 Current software architecture ..................................................................................37
  6.3 Proposed software architecture ...............................................................................37
    6.3.1 Overview ..............................................................................................................37
    6.3.2 Subsystem decomposition ...................................................................................39
    6.3.3 Hardware/software mapping ..............................................................................39
    6.3.4 Access control and security ..............................................................................40
  6.4 User interface ............................................................................................................40
7 Test plans ......................................................................................................................41
  7.1 Features to be tested .................................................................................................41
  7.2 Pass/fail criteria .........................................................................................................42
  7.3 Approach ..................................................................................................................43
  7.4 Test cases ..................................................................................................................43
8 Project issues .................................................................................................................43
  8.1 Open issues ...............................................................................................................43
  8.2 Potential user problems ............................................................................................44
  8.3 Costs ..........................................................................................................................44
  8.4 Waiting room ............................................................................................................45
  8.5 Project retrospective ...............................................................................................46
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High level view of system</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Current hosting system's dashboard</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Current hosting system's detail</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>Door's data table diagram</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>User's data table diagram</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Ratio of active users after signup</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>User Events</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>System's Daily Active Installation</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>Application's Drawer Options</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>Application's HomePage</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>Application's Signup</td>
<td>18</td>
</tr>
<tr>
<td>12</td>
<td>Application's login</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>Application's Message Thread</td>
<td>19</td>
</tr>
<tr>
<td>14</td>
<td>System Fetching Messages from server</td>
<td>19</td>
</tr>
<tr>
<td>15</td>
<td>Application's Message Archive</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>Current System's Use Case Diagram</td>
<td>26</td>
</tr>
<tr>
<td>17</td>
<td>Current System's Sequence Diagram</td>
<td>27</td>
</tr>
<tr>
<td>18</td>
<td>Current System's Data Diagram</td>
<td>28</td>
</tr>
<tr>
<td>19</td>
<td>User Schema Data Diagram</td>
<td>29</td>
</tr>
<tr>
<td>20</td>
<td>Door Schema Data Diagram</td>
<td>29</td>
</tr>
<tr>
<td>21</td>
<td>User Schema Data Diagram</td>
<td>30</td>
</tr>
<tr>
<td>22</td>
<td>Current System Architecture</td>
<td>38</td>
</tr>
<tr>
<td>23</td>
<td>Proposed Architecture</td>
<td>39</td>
</tr>
<tr>
<td>24</td>
<td>Current System's Component Diagram</td>
<td>40</td>
</tr>
<tr>
<td>25</td>
<td>Color Pallet Evaluation</td>
<td>41</td>
</tr>
<tr>
<td>26</td>
<td>Final Application Theme</td>
<td>42</td>
</tr>
<tr>
<td>27</td>
<td>Test Case 1</td>
<td>43</td>
</tr>
<tr>
<td>28</td>
<td>Test Case 2</td>
<td>43</td>
</tr>
<tr>
<td>29</td>
<td>Prospective login process</td>
<td>46</td>
</tr>
</tbody>
</table>
3 LIST OF TABLES

Table 1. Goal Measurement Hypotheses........................................................................................................9
Table 2. List of Product Scenarios..................................................................................................................17
Table 3. User Priorities ....................................................................................................................................20
Table 4. User Participation..............................................................................................................................21
Table 5. Millgun Track Events........................................................................................................................22
Table 6. Anticipated Workplace Environment Attributes................................................................................23
Table 7. Project's open issues..........................................................................................................................43
Table 8. Prospective Features..........................................................................................................................45
4 PROJECT DESCRIPTION

4.1 PROJECT OVERVIEW

Normally university professors are not present at all times in their office, even during work hours, and sometimes during the office owner’s absence, students and other people visit and don’t find the professor. This situation causes problems in student-professor communication. Therefore, we address the problem of asynchronous cooperation between students and professors. We have proposed a cloud-assisted Internet of Things (IoT) based Cyber-Physical door to boost student-professor interaction in the times of absence of the professor at his room. By this system, professor and students can interact virtually by transferring messages among themselves [1].

4.2 THE PURPOSE OF THE PROJECT

Ubiquitous computing, pervasive computing, Internet Protocol, sensing technologies, communication technologies, and embedded devices are merged together in order to form a system where the real and digital worlds meet and are continuously in symbiotic interaction. The smart object is the building block of the IoT vision. By putting intelligence into everyday objects, they are turned into smart objects able not only to collect information from the environment and interact with or control the physical world, but also to be interconnected through Internet to exchange data and information. The expected huge number of interconnected devices and the significant amount of available data open new opportunities to create services that will bring tangible benefits to the university professors and students to interconnect virtually [2].

4.2.1 THE USER BUSINESS OR BACKGROUND OF THE PROJECT EFFORT

When you look for a professor at his office and he is not there, it is common at a workplace to leave a message on a little whiteboard or some paper outside the office. However, this approach has some security and privacy issues. Another alternative would be a door-messaging machine that allows people to record a message that would be sent to the office owner. We have been motivated by the problem addressed above to propose a cloud assisted IoT based social door to boost student-professor interaction. In our approach, we intend to use NFC and QR-code tags to make a unique virtual identity each door. Virtual identity will inset the door in the server and can ease the communication between students and the door in the absence of the professor. Each door will be aware of its owner and the surrounding environment, for instance information like students who intended to have a meeting with professor can be reported to the professor himself, which would not be at hand before. By this approach, we expect to resolve problems of communicating with the professor in his absence and to increase the subjective satisfaction of student-professor interaction. It is expected that professors and students use the delivered product to exchange messages while the professor is not physically present at his office. This system can be thought as a virtual secretary for the professor [3].

4.2.2 THE CONTEXT OF THE WORK

In order to better illustrate our proposed system, we created a context diagram of the system to define the boundary between the system, or part of a system, and its environment, showing the entities that interact with it. Figure 1 illustrates a high level view of the system.
Cloud-assisted IoT based Social Door to Boost Student-Professor Relationship

**Figure 1.** High level view of system

Users login to identify themselves.

Using NFC or QR, mobile application determines door ID and location and start communicating with the cloud-based server.

After connecting to the server, the system checks the users' access level and identity.

Users submit their presence or reply to professor's message if permission was given by the professor in advance.

Regardless of which technology was previously used, professor's message is fetched from the server and is shown on the screen. Users can perform specific actions based on their access level.
4.2.3 GOALS OF THE PROJECT

We want to enhance the subjective satisfaction of the student-professor interaction. We will provide professors and students with the feasibility to interact with each other virtually while the professor is not available at his office.

4.2.4 MEASUREMENT

Our goal to boost the subjective satisfaction in student-professor interaction is expected to bring convenience for both students and professors. Table 1 indicates our goal measurements hypotheses and the benefits we believe our system will produce for our target group.

<table>
<thead>
<tr>
<th>Hypotheses</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[H1] Increase in student’s engagement in course projects.</td>
<td>Students will feel more motivated to participate in course projects when they observe a live pilot of a project under the professor’s supervision.</td>
</tr>
<tr>
<td>[H2] Increase in student's request to participate in lab groups.</td>
<td>Students can make sure the professor will track their work when they observe how professor allocates his consideration to keep track of every student trying to contact him.</td>
</tr>
<tr>
<td>[H3] Improvement in trust between professor and students</td>
<td>By this system professor can make sure of the students’ presence at his doorstep and students can also provide professor with proof of their activities which can be expected to boost trust in student-professor relationship.</td>
</tr>
<tr>
<td>[H4] Increase the availability of the professor</td>
<td>Professor can always leave messages for his student even if he is not physically available at his work office. By this approach the system can benefit the professor-student interaction as a virtual secretary.</td>
</tr>
<tr>
<td>[H5] Improvement in Information flow between students</td>
<td>Same sources of messages of the professor will be available for all students. Due to the student’s eagerness to communicate with the professor and to meet him in person, they can have access to the information even in the absence of the professor.</td>
</tr>
<tr>
<td>[H6] Increase in Security of the information flow</td>
<td>All students will have well defined authentication strategies and very single task they do will be monitored in the server. This procedure will enhance the security of information flow between students and professors</td>
</tr>
<tr>
<td>[H7] Lower the response time of the professor</td>
<td>Professor will be informed of his students’ activity by email and will be available to respond to the email if he wants to. This strategy will lower the response time of the professor and therefore increase the satisfaction of the users.</td>
</tr>
</tbody>
</table>
4.3 THE SCOPE OF THE WORK

As in any software engineering process, the first stage of augmented objects developing process deals with the problem space. This stage identifies restrictions and opportunities that can be exploited developing a computer system that could use one or more augmented object as interface. Once the problem is defined, a set of requirements is gathered and the context of use is established. After the main requirements are gathered, the development process begins. Because the interface is not going to be graphical (i.e., not a graphical user interface), the next step is the definition of the object or objects that would allow the interaction with the system. Finally, as in traditional software engineering processes, a testing phase is needed in order to verify and validate in the final product [4].

4.3.1 THE CURRENT SITUATION

We have implemented an Android application that students use it to communicate with smart door. Each door has a unique identity which is saved in QrCode and NFC tags installed on the door. The project’s server side uses Parse and its libraries. Students can use the Android application to login to their smart door accounts via email and read professor messages and to submit their presence. Professor will use the Parse server side to upload his messages for students. Messages can be specified for custom group of students with diverse access rights which can be defined on the server.

We used Bitbucket as our version control hosting system. Figure 2 shows the hosting system’s dashboard.

Figure 2. Current hosting system's dashboard
The hosting version control system can be accessed here with “smartdoor_ami” as username and password Value as the password. Figure 3 represent some detail information about this host. This host was built with “smartdoor.maillog@gmail.com” email address and under the name and authorization of AmIIab as the owner company.

In the current version of the system, we use Parse for storing our data. Each door will be defined there with their access levels and containing messages. Students will also be stored in another table with their specific access levels which will be used to authenticate their permission to access the door’s message. Figure 4 shows the door table on the server and Figure 5 indicates the users’ table. Based on the server’s statistics 20% of users were active after installing and signing up for the first time to the application. Figure 6 represents the server’s report of ratio of active users after days of signing up to the system.

---

1 Password: asdf;lkjgqerpoiu
Figure 4. Door's data table diagram

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Door ID</td>
<td>12345</td>
</tr>
<tr>
<td>Location</td>
<td>Room A</td>
</tr>
<tr>
<td>Status</td>
<td>Open</td>
</tr>
<tr>
<td>Date</td>
<td>2023-04-01</td>
</tr>
</tbody>
</table>

This table provides data related to the door's status and location.
Figure 5. User's data table diagram
As we released our first version to the HCI course students and Ambient Intelligence Laboratory members, we have 16 active users now to perform our alpha testing and to evaluate users’ expectations and mental model form the system. Figure 7 represents the events our users have triggered up until now.

In addition to the AmI Lab members and HCI course students, other graduate and undergraduate students have requested to access the application. With the confirmation of the owner (Dr. Ali Asghar Nazari Shirehjini) we released the application to the students. All requests and responds and potential errors and question were reported and answered to the supporting email address of the system. Figure 8 shows the daily active installations of the application and Figure 9 represents system’s daily active users.

### 4.4 COMPETING PRODUCTS

Many of the studies in the area of ubiquitous computing and augmented objects are strictly related with the use and communication of the objects but not with the development process itself. In most of those studies the topics of smart objects, the Internet of Things, and Pervasive Computing are being addressed, but most of them are presenting frameworks or solutions to specific problems found in the development of the objects. Our research differs because it focuses on the creation of augmented objects as a process with a real problem as the starting point [5].

---

2 smartdoor.maillog@gmail.com
A straightforward way that authors have used to augment objects without the necessity of embedding computational capabilities in them is using communication technologies, for instance, Bravo, Chavira, Nava, and Villareal (2008) compared RFID and NFC technologies when augmenting environments in an Ambient Intelligence (AmI) context. Because objects necessarily must have an identity, these comparisons are very useful in deciding which technology to use.

Vega, Casado, and Valero (2012) presented a method to standardize the communication of things through the Internet to allow the creation of smart spaces [6].

Most of the revised literature discusses the particular characteristics that an augmented object should have in a particular domain. However, we did not find works related to the design and implementation of augmented objects from a Software Engineering perspective.

---

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Dimension(s)</th>
<th>Installation ID</th>
<th>Parse User ID</th>
<th>Parse SDK Version</th>
<th>OS</th>
<th>OS Version</th>
<th>App Build Version</th>
<th>App Display Version</th>
<th>Timestamp (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubmitPresence</td>
<td>&quot;approach&quot;, &quot;day1&quot;, &quot;weekend&quot;, &quot;mon&quot;</td>
<td>63670707-ee37-422e-96f1-7ec774df1ed</td>
<td>9bf2b350f1</td>
<td>Android</td>
<td>1.18.0</td>
<td>Android</td>
<td>5.0.2</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>GetMessage</td>
<td>&quot;approach&quot;, &quot;day1&quot;, &quot;weekend&quot;, &quot;mon&quot;</td>
<td>63670707-ee37-422e-96f1-7ec774df1ed</td>
<td>9bf2b350f1</td>
<td>Android</td>
<td>1.18.0</td>
<td>Android</td>
<td>5.0.2</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>AppOpened</td>
<td>{}</td>
<td>63670707-ee37-422e-96f1-7ec774df1ed</td>
<td>9bf2b350f1</td>
<td>Android</td>
<td>1.18.0</td>
<td>Android</td>
<td>5.0.2</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>AppOpened</td>
<td>{}</td>
<td>782bf6d9-179d-48a8-a57e-b235dc4df8</td>
<td>0tSTDcJQwp</td>
<td>Android</td>
<td>1.18.0</td>
<td>Android</td>
<td>5.0.2</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>AppOpened</td>
<td>{}</td>
<td>782bf6d9-179d-48a8-a57e-b235dc4df8</td>
<td>0tSTDcJQwp</td>
<td>Android</td>
<td>1.18.0</td>
<td>Android</td>
<td>5.0.2</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>ReplyToMessage</td>
<td>&quot;approach&quot;, &quot;day1&quot;, &quot;weekend&quot;, &quot;Sun&quot;</td>
<td>782bf6d9-179d-48a8-a57e-b235dc4df8</td>
<td>0tSTDcJQwp</td>
<td>Android</td>
<td>1.18.0</td>
<td>Android</td>
<td>5.0.2</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>GetMessage</td>
<td>&quot;approach&quot;, &quot;day1&quot;, &quot;weekend&quot;, &quot;Sun&quot;</td>
<td>782bf6d9-179d-48a8-a57e-b235dc4df8</td>
<td>0tSTDcJQwp</td>
<td>Android</td>
<td>1.18.0</td>
<td>Android</td>
<td>5.0.2</td>
<td>1</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Figure 7. User Events
4.5 PRODUCT SCENARIOS

User can the use the released Android mobile application for submitting their presence, reading professor’s messages, surveying through archived messages and replying to professor’s message.
4.5.1 PRODUCT SCENARIO LIST

Table 2 represents the scenarios that can be triggered by users via the android mobile application, person who can trigger the mentioned scenarios.

Table 2. List of Product Scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Applicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signup</td>
<td>Student</td>
<td>Students can request for creating an account for a door by signing up in the Android mobile application.</td>
</tr>
<tr>
<td>Login</td>
<td>Student, Professor</td>
<td>Students and Professors can use their Android mobile application to login to their smart door accounts using a username and password they created in signup section.</td>
</tr>
<tr>
<td>Submit Presence</td>
<td>Student</td>
<td>Students can submit their presence via the Android mobile application to inform the professor of their intent to meet with him.</td>
</tr>
<tr>
<td>Reply Message</td>
<td>Student, Professor</td>
<td>Students can reply to professor’s messages if they are granted by the reply access right by the professor.</td>
</tr>
<tr>
<td>Survey through archived messages</td>
<td>Student, Professor</td>
<td>Students and Professors can use their Android mobile applications to have a survey through their previous conversations and their exchanged messages.</td>
</tr>
</tbody>
</table>

4.5.2 INDIVIDUAL PRODUCT SCENARIOS

- After signing up for smart door application account, users will be directed to login page. They can use the username and password they created on the signup section to get access to their accounts. Figure 12 shows the login page of the application.

- To use the smart door project one must have an account. Students can use the signup page of the Android mobile application to create a smart door account. Their username and password will be defined by themselves and will be used in prospective usage of the system. Figure 11 shows the signup page of the application.

- If inserted correct information from users in the login page, users will be directed to the homepage. Figure 10 represents the home page of the application.

Students can use one of the two approaches of reading a QRCode or a NFC tag to read the door's identity and to connect to server to get the professor's message.

- An Android drawer is placed at the top left corner of the application to ease the navigation through each page. Figure 9 shows the different options of the drawer.

- Despite of the technology being used for reading door's identity, after the application gets access to door's id, it will send a request to the server. The server will answer to this
request with message or access updates. Users will have the option to reply to the message or submit their presence of their access is granted by the professor. Figure 14 expresses this scenario.
• Users are able to survey through archived exchanged messages among themselves. Figure 13 shows the message thread of the application.

• By clicking on the message thread, messages exchanged between users will be shown. Figure 15 indicates the archived messages.

4.6 STAKEHOLDERS

As mentioned before, smart door project is specifically designed and implemented for academic environments and to boost student-professor interaction in the absence of the professor at his office room. Students and professors are mentioned as the main stakeholders of this project since they will benefit the most from the project’s usage.

4.6.1 THE STUDENT

The students as the clients of the system have the final say on acceptance of the product, and thus must be satisfied with the product as delivered. Students can be thought of the client as the person who makes the investment in the product. Since the students vary from 1st year seniors to Lab members and PhD students, numerous details had to be considered. All of these groups’ expectations must be found out and be considered for the final release of the product.

4.6.2 THE PROFESSOR

Professor can be thought as the customer of the system. Professor may intend to use this system in order to boost their interaction while their not present at their office room. They might also want to use other features of the product in their laboratories. For instance this product can be used for making sure of the lab member’s presence since the professor will not make time to check on them simultaneously. Professor will be able to access a report of all users whom are assigned to his door.
Due to the ratio of the need of the users to this system and how critical this system can be for solving their interaction, we have classified the users into three groups. Table 3 shows different group of the users and the priority of them.

Table 3. User Priorities

<table>
<thead>
<tr>
<th>User</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Users</td>
<td>University Professors who have independent laboratories or office rooms, undergraduate students whom don’t have the opportunity to meet the professor in person.</td>
</tr>
<tr>
<td>Secondary Users</td>
<td>Teaching Professors who don’t have independent laboratories or office rooms, PhD and graduate students who work in laboratories under the target professor’s supervision and will have personal meetings with the professor on a daily schedule.</td>
</tr>
<tr>
<td>Unimportant Users</td>
<td>Staff members.</td>
</tr>
</tbody>
</table>

4.6.3 USER PARTICIPATION

Participation of the users listed below is expected. Their role in the system is also mentioned in the table below.
Table 4. User Participation

<table>
<thead>
<tr>
<th>Key Users</th>
<th>University Professors who have independent laboratories or office rooms, undergraduate students whom don’t have the opportunity to meet the professor in person.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secondary Users</td>
<td>Teaching Professors who don’t have independent laboratories or office rooms, PhD and graduate students who work in laboratories under the target professor’s supervision and will have personal meetings with the professor on a daily schedule.</td>
</tr>
<tr>
<td>Unimportant Users</td>
<td>Staff members.</td>
</tr>
</tbody>
</table>

4.7 MANDATED CONSTRAINTS

During the eventual design of the product we faced some mandated constraints that users must meet the minimum requirements in order to use the application. The list below expresses these constraints.

1. Users must have Android smart phones in order to use the system.
2. Internet Connection is required for the system usage.
3. Users’ smart phones must be able to read NFC tags or read Qrcodes.

4.7.1 SOLUTION CONSTRAINTS

We introduce solutions to the constraints listed above.

1. Description: Users must have Android smart phones in order to use the system.

   Rationale: Users must have a tool as a user interface to interact with the system’s server.
   Fit criterion: Since most of the university students have android smart phones, and android mobile application was implemented to be used as the client side interface to interact with the system.

2. Description: Internet Connection is required for the system usage.

   Rationale: Since our augmented door does not have processing capacities, an interconnection third party must be used in order to receive users’ request and respond to them.
   Fit criterion: Since the target environment is academic universities, it’s expected from these environments to provide internet connections to use in the system.

3. Description: Users’ smart phones must be able to read NFC tags or read Qrcodes.

   Rationale: Doors’ identity must be in system’s hand to perform well.
   Fit criterion: In addition to the ability of reading NFC tags most of the Android smart phones, we used a QrCode reader library to read doors’ identity from QrCode tags.
4.7.2 IMPLEMENTATION ENVIRONMENT OF THE CURRENT SYSTEM

4.7.2.1 Cloud:

We used Parse as our cloud server. Parse's vision is to let developers build any mobile app without dealing with servers. For complex apps, sometimes you just need a bit of logic that isn't running on a mobile device. Cloud Code makes this possible. Cloud Code is easy to use because it's built on the same JavaScript SDK that powers thousands of apps. The only difference is that this code runs in the Parse Cloud rather than running on a mobile device. When you update your Cloud Code, it becomes available to all mobile environments instantly. You don't have to wait for a new release of your application. This lets you change app behavior on the fly and add new features faster [7].

4.7.2.2 Android Backend:

The Parse platform provides a complete backend solution for your mobile application. Our goal is to totally eliminate the need for writing server code or maintaining servers. If you're familiar with web frameworks like Ruby on Rails, we've taken many of the same principles and applied them to our platform. In particular, Parse's SDK is ready to use out of the box with minimal configuration on client part.

On Parse, you create an App for each of your mobile applications. Each App has its own application id and client key that you apply to your SDK install. Your account on Parse can accommodate multiple Apps. This is useful even if you have one application, since you can deploy different versions for test and production [8].

4.7.3 PARTNER OR COLLABORATIVE APPLICATIONS

In addition to using Parse as our cloud server and using Android backend for implementing an interface to interact with the system. We used Mailgun for tracking and reporting students’ activities to the professor. Mailgun keeps track of every event that happens to every message (both inbound and outbound) and stores this data for at least 30 days for paid accounts and 2 days for free accounts. Below is the table of events that Mailgun tracks [9].

Table 5. Mailgun Track Events

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accepted</td>
<td>Mailgun accepted the request to send/forward the email and the message has been placed in queue.</td>
</tr>
<tr>
<td>Delivered</td>
<td>Mailgun sent the email and it was accepted by the recipient email server.</td>
</tr>
<tr>
<td>Failed</td>
<td>Mailgun could not deliver the email to the recipient email server.</td>
</tr>
<tr>
<td>Opened</td>
<td>The email recipient opened the email and enabled image viewing. Open tracking must be enabled in the Mailgun control panel, and the CNAME record must be pointing to mailgun.org.</td>
</tr>
</tbody>
</table>
The email recipient clicked on a link in the email. Click tracking must be enabled in the Mailgun control panel, and the CNAME record must be pointing to mailgun.org.

The email recipient clicked on the unsubscribe link. Unsubscribe tracking must be enabled in the Mailgun control panel.

The email recipient clicked on the spam complaint button within their email client. Feedback loops enable the notification to be received by Mailgun.

Mailgun has stored an incoming message.

<table>
<thead>
<tr>
<th>Clicked</th>
<th>The email recipient clicked on a link in the email. Click tracking must be enabled in the Mailgun control panel, and the CNAME record must be pointing to mailgun.org.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsubscribed</td>
<td>The email recipient clicked on the unsubscribe link. Unsubscribe tracking must be enabled in the Mailgun control panel.</td>
</tr>
<tr>
<td>Complained</td>
<td>The email recipient clicked on the spam complaint button within their email client. Feedback loops enable the notification to be received by Mailgun.</td>
</tr>
<tr>
<td>Stored</td>
<td>Mailgun has stored an incoming message.</td>
</tr>
</tbody>
</table>

### 4.7.4 ANTICIPATED WORKPLACE ENVIRONMENT

Workplace environments are academic university departments which both users and students work in. These environments are believed to have several attributes that are listed in the table below.

`Table 6. Anticipated Workplace Environment Attributes`

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>light</td>
<td>The lightening of the environment must be in a standard level. In order to provide the possibility of reading QRCode tags with smartphone’s camera.</td>
</tr>
<tr>
<td>Internet Connection</td>
<td>As mentioned before, Internet connection is crucial for interconnecting the client Android application side with the processing cloud server.</td>
</tr>
</tbody>
</table>

### 4.8 NAMING CONVENTIONS AND DEFINITIONS

#### 4.8.1 DEFINITIONS OF KEY TERMS

- **Internet of Things (IoT):**

  The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of almost 50 billion objects by 2020.
• Augmented Object:

There are many examples of augmented objects in the literature. Augmented objects should provide intelligence to the ambient where they are located and also they must require a low cognitive effort to be used. Few works have been reported providing guidelines to conceive and design these components. Therefore, developers have to use improvised ad hoc software processes to support the development of augmented objects. In order to help dealing with this situation, this article presents a software process to develop these components. The proposed process was named Augmented Objects Development Process (AODEP) and it is based on the authors’ previous experiences and software engineering best practices. The article also reports two case studies in which AODEP was used to guide the development of augmented objects in specific problems. The obtained results are encouraging.

• Cloud Assisted:

A cloud server is a logical server that is built, hosted and delivered through a cloud computing platform over the Internet. Cloud servers possess and exhibit similar capabilities and functionality to a typical server but are accessed remotely from a cloud service provider. A cloud server may also be called a virtual server or virtual private server. A cloud server is primarily an Infrastructure as a Service (IaaS) based cloud service model. There are two types of cloud server: logical and physical. A cloud server is considered to be logical when it is delivered through server virtualization. In this delivery model, the physical server is logically distributed into two or more logical servers, each of which has a separate OS, user interface and apps, although they share physical components from the underlying physical server. Whereas the physical cloud server is also accessed through the Internet remotely, it isn’t shared or distributed. This is commonly known as a dedicated cloud server.

• Cyber-Physical Systems (CPS):

Due to the huge advancements in the fields of electronics and the deployments of wireless communication systems, mobile devices and ubiquitous services (providing anytime-anywhere connectivity to the users) spread rapidly over the past decade. Today, however, the role played by devices is no longer limited to connect users to the Internet, but it has been expanding becoming an opportunity to interlink the physical world with the cyber world, leading to the emergence of Cyber-Physical Systems (CPS). The notion of CPS refers to a next generation of embedded ICT systems where computation and networking are integrated with physical processes and they control and manage their dynamics and make them more efficient, reliable, and adaptable and secure.

4.8.2 UML AND OTHER NOTATION USED IN THIS DOCUMENT

This document generally follows the Version 2.0 OMG UML standard, as described by Fowler in [4].

4.9 RELEVANT FACTS AND ASSUMPTIONS

During the implementation of the system some factors that have an effect on the product, but are not mandated requirements constraints (e.g. business rules, organizational systems, or any other
activities that have an effect on this product) were considered and some assumptions have been made for this version of the system.

4.9.1 FACTS

The list below expresses the factors about the current implementation version of the system:

1. Users need Android Smart phones to use the system.
2. The anticipated workplace must provide internet connection in order the provide the interaction of server and client side
3. Existence of QRCode and NFC tags are necessary for system usage.

4.9.2 ASSUMPTIONS

The list below expresses the assumptions about the prospective implementation version of the system:

1. The iOS version of the client interface will be implemented for the prospective versions of the system in order to allow iPhone users to benefit this system.
2. A specific mobile application will be implemented for the professor so that he would be able to upload his messages to the door using his smart phone.
3. New NFC tags would be used in prospective system version that will support the file transfer ability. This feature can provide the possibility of file sharing between professor and students.

5 REQUIREMENTS

Starting point for deciding if the problem or part of the problem can be addressed using AO. If we decide to use AO as part of the solution, we suggest using observations, diary studies, and concept validation techniques to better define the user requirements. Once the requirements are clear, a metaphor can be used to identify the best candidate objects.

5.1 PRODUCT USE CASES

In this section we will indicate our system’s sequence of actions that provide a measurable value to an actor. Another way to look at it is a use case describes a way in which a real-world actor interacts with the system. In a system use case you include high-level implementation decisions.

5.1.1 USE CASE DIAGRAMS

Figure 16 expresses our system’s use case. User, student and professor entities has been defined as our system’s actors. The term “user” is allocated to a system user whom doesn’t have any defined role in the system (e.g. first time user or someone who passes by the system and has the application installed on his smart phone and wants to give the system a try).
Figure 16. Current System's Use Case Diagram
5.2 FUNCTIONAL REQUIREMENTS

In this system, users must login to their smart door accounts to be able to use the system’s facilities. After users’ authentication, they will be able to submit their presence, have access to professor’s messages and to reply to the message. Each user will only have access to his own messages and information. And the professor will be able to track the student’s activities and to give them specific access levels. Figure 17 shows the sequence diagram of the project.

![Sequence Diagram](image)

*Figure 17. Current System's Sequence Diagram*

5.3 DATA REQUIREMENTS

For current version of the system, we used Json objects to store the data in our server, Figure 18 shows current data decomposition in server.
Figure 18. Current System's Data Diagram
Figure 19 shows the user schema while Figure 20 shows the door schema of the system. Figure 21 shows our user data schema in our current version of the system. Each user has a defined role by the door owner and can have multiple access to get access to the messages stored in the cloud server. Users do also have the attributes of “canReply” and “lastPresence” to identify their authentication of being able to reply to professor’s messages and their last presence by the door. Users’ email values, sessions, name, type, encrypted password and their entity type is also stored on the cloud server. Users’ entity type declares their role in the system and it can vary to “professor”, “student”, “null” and “public”. In the current version of the system public and null entity types are equal because they both don’t have any specific type define for them by the door owner.
Figure 21. User Schema Data Diagram
5.4 PERFORMANCE REQUIREMENTS

Users expect to send their request and receive the professors’ answers quickly. System should also provide the users with proper feedbacks with potential errors. For instance system must inform the users to turn on their NFC connection or to connect to Internet whenever necessary.

5.4.1 SPEED AND LATENCY REQUIREMENTS

Messages should not be fetched and shown to the user more than a specific amount of time or it will become frustrating for the users to use this system for communicating with the professor.

5.4.2 PRECISION OR ACCURACY

Since the messages are stored as Json objects on server and they will be fetched directly from the Parse cloud server, there are no accuracy concerns about this aspect for now. In addition, some human errors might accrue from the professor when entering new access levels on the server for specific group of students. Nevertheless in prospective versions of the system a specific Android mobile application will be ready to use as an interface for the professor to interact with the system.

5.4.3 CAPACITY REQUIREMENTS

In current version of the system, we only store the last 20 messages exchanged between the professor and students. In prospective versions we intend to contain expandable scroll list to fetch every message ever exchanged between the user and the professor.

5.5 DEPENDABILITY REQUIREMENTS

5.5.1 RELIABILITY REQUIREMENTS

Since the system’s availability and performance is only relied on the internet connection and server’s availability, it would be reliable under specific requirements. Our current cloud server is available for 30 user’s request per second.

5.5.2 AVAILABILITY REQUIREMENTS

System’s availability highly relies on the availability to internet connection and server existence. Since augmented doors do not have processing abilities, it is crucial that client side Android application interface be able to send request to the server and to fetch the answers back to show the results to the user.

5.5.3 ROBUSTNESS OR FAULT-TOLERANCE REQUIREMENTS

We have developed our system to be complete robust. The client side interface for interacting with the server is completely fault tolerant and will give feedbacks to users if any error happens while interconnecting with the server or showing the response to users’ request.

5.5.4 SAFETY-CRITICAL REQUIREMENTS

There are no safety-critical requirements implementing or using our project. All steps to re-create or to use the project are completely safe and need no safety-critical requirements.
5.6 MAINTAINABILITY AND SUPPORTABILITY REQUIREMENTS

5.6.1 MAINTENANCE REQUIREMENTS

This system requires maintenance in six sections expressed below:

1. The implementation process contains software preparation and transition activities, such as the conception and creation of the maintenance plan; the preparation for handling problems identified during development; and the follow-up on product configuration management.

2. The problem and modification analysis process, which is executed once the application has become the responsibility of the maintenance group. The maintenance programmer must analyze each request, confirm it (by reproducing the situation) and check its validity, investigate it and propose a solution, document the request and the solution proposal, and finally, obtain all the required authorizations to apply the modifications.

3. The process considering the implementation of the modification itself.

4. The process acceptance of the modification, by confirming the modified work with the individual who submitted the request in order to make sure the modification provided a solution.

5. The migration process (platform migration, for example) is exceptional, and is not part of daily maintenance tasks. If the software must be ported to another platform without any change in functionality, this process will be used and a maintenance project team is likely to be assigned to this task.

6. Finally, the last maintenance process, also an event which does not occur on a daily basis, is the retirement of a piece of software.

5.6.2 SUPPORTABILITY REQUIREMENTS

Our system needs a cloud server for storing message and responding to users’ requests. In addition we intend to use a web server to support activity notification for informing the professor. Currently we use the Parse as our cloud server and Mailgun as our email notify service. The system is highly relied on these two services.

5.6.3 ADAPTABILITY REQUIREMENTS

Our system is designed to be highly adaptive and to be used for other environments rather than academic universities. Corporate environments which can provide the internet connection or local servers are potential nominees to use the current version of the system.

5.6.4 SCALABILITY REQUIREMENTS

The current free version of our server supports 30 requests per seconds for now. With improvement in quantity of the users, the server has to be updated. The current service of parse can handle 600 request per seconds with 5700$ per month.

5.6.5 LONGEVITY REQUIREMENTS

This system is believed to be alive as far as students and professors intend to use it. In addition the system is highly relied on the cloud server. With the existence of the server, the only crucial matter for the system’s longevity is the users’ participation.
5.7 SECURITY REQUIREMENTS

5.7.1 ACCESS REQUIREMENTS

Users will be authenticated by their smart phone accounts and each of their activities would be tractable by the professor. Professor himself will provide other users with diverse access levels and is able to change the access rights when necessary.

5.7.2 INTEGRITY REQUIREMENTS

Aspects below should be considered for the system integrity:
1. That condition of a system wherein its mandated operational and technical parameters are within the prescribed limits.
2. The quality of an AIS when it performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.
3. The state that exists when there is complete assurance that under all conditions an IT system is based on the logical correctness and reliability of the operating system, the logical completeness of the hardware and software that implement the protection mechanisms, and data integrity.

5.7.3 PRIVACY REQUIREMENTS

Users expect that their personal information and interactions with the system be private and not accessible by other users. In our system each users is only allowed to have access to his own information and exchanged messages with the professor and only the professor is allowed to see each of his student’s activity and messages.

5.8 USABILITY AND HUMANITY REQUIREMENTS

5.8.1 EASE OF USE REQUIREMENTS

Since most of people have smart phones with them most of the times with them, implementing an Android based system is thought to be convenient to use and accessible at any time anywhere.

5.8.2 PERSONALIZATION AND INTERNATIONALIZATION REQUIREMENTS

Since the system is aimed for use in academic universities, it’s expected that university students be familiar with English, therfore this system is implemented in English language. Nevertheless, all messages exchanges between professor and students can be written in any customized language that Android operating system supports.

5.8.3 LEARNING REQUIREMENTS

This system is implemented with Android mobile application as the interface for communicating with the system. In addition to the assumption that students and professors are familiar with mobile applications, a friendly user interface has been created to boost the user-system interaction. There is also an introduction section of the application ready for users. Therefore use of this system needs no special learning requirements for end users.
5.8.4  UNDERSTANDABILITY AND POLITENESS REQUIREMENTS

While designing the interface of the application, we tried to use both Android standard symbols and icons and symbols that are expected that users would be familiar with them in order to enhance the understandability of the system for users and to express the system’s functionality.

5.8.5  USER DOCUMENTATION REQUIREMENTS

In addition to the friendly user interface of mobile applications to boost the participation of the users in the system, a user manual is also prepared to show users step by step how each section of the application will work, what actions are necessary to use the system properly and how their request will be respond using the system. Section 4.4.2 also indicates all scenarios that user will face using this system.

5.8.6  TRAINING REQUIREMENTS

As explained in 5.8.3 section, there is no training or learning experience needed to use the system for end users.

5.9  LOOK AND FEEL REQUIREMENTS

5.9.1  APPEARANCE REQUIREMENTS

Since the users of this system include both professors and university students who are senior adults, we tried our best to both use technologies that are attractive and easy to learn and use. We used Android Material design principals. We challenged ourselves to create a visual language for our users that synthesizes the classic principles of good design with the innovation and possibility of technology and science. This is material design. This spec is a living document that will be updated as we continue to develop the tenets and specifics of material design.

Our goals to use this design was to:

- Create a visual language that synthesizes classic principles of good design with the innovation and possibility of technology and science.
- Develop a single underlying system that allows for a unified experience across platforms and device sizes. Mobile precepts are fundamental, but touch, voice, mouse, and keyboard are all first-class input methods.

5.9.2  STYLE REQUIREMENTS

- The product shall appear authoritative.
- The foundational elements of print-based design – typography, grids, space, scale, color, and use of imagery – guide visual treatments. These elements do far more than please the eye. They create hierarchy, meaning, and focus. Deliberate color choices, edge-to-edge imagery, large-scale typography, and intentional white space create a bold and graphic interface that immerse the user in the experience.
- An emphasis on user actions makes core functionality immediately apparent and provides waypoints for the user.
- Motion respects and reinforces the user as the prime mover. Primary user actions are inflection points that initiate motion, transforming the whole design.
- All action takes place in a single environment. Objects are presented to the user without breaking the continuity of experience even as they transform and reorganize.
• Motion is meaningful and appropriate, serving to focus attention and maintain continuity. Feedback is subtle yet clear. Transitions are efficient yet coherent.

5.10 OPERATIONAL AND ENVIRONMENTAL REQUIREMENTS

5.10.1 EXPECTED PHYSICAL ENVIRONMENT

The product must be used by users with Android smart phones that have NFC sensors or can read QRCode tags and have the access to Internet connection and
Requirements for Interfacing with Adjacent Systems
This system will perfectly run on smart phones with Android operating systems higher than 4.0.3 version.

5.10.2 PRODUCTIZATION REQUIREMENTS

Preparing your application for release is a multi-step process that involves the following tasks:
• Configuring your application for release.
  At a minimum you need to remove Log calls and remove the android:debuggable attribute from your manifest file. You should also provide values for the android:versionCode and android:versionName attributes, which are located in the <manifest> element. You may also have to configure several other settings to meet Google Play requirements or accommodate whatever method you’re using to release your application.
• If you are using Gradle build files, you can use the release build type to set your build settings for the published version of your app.
• Building and signing a release version of your application.
  You can use the Gradle build files with the release build type to build and sign a release version of your application. See Building and Running from Android Studio.
• Testing the release version of your application.
  Before you distribute your application, you should thoroughly test the release version on at least one target handset device and one target tablet device.
• Updating application resources for release.
  You need to be sure that all application resources such as multimedia files and graphics are updated and included with your application or staged on the proper production servers.
• Preparing remote servers and services that your application depends on.
  If your application depends on external servers or services, you need to be sure they are secure and production ready.

5.10.3 RELEASE REQUIREMENTS

Google Play is a robust publishing platform that helps you publicize, sell, and distribute your Android applications to users around the world. When you release your applications through Google Play you have access to a suite of developer tools that let you analyze your sales, identify market trends, and control who your applications are being distributed to. You also have access to several revenue-enhancing features such as in-app billing and application licensing. The rich array of tools and features, coupled with numerous end-user community features, makes Google Play the premier marketplace for selling and buying Android applications.

Releasing your application on Google Play is a simple process that involves three basic steps:
• Preparing promotional materials.
To fully leverage the marketing and publicity capabilities of Google Play, you need to create promotional materials for your application, such as screenshots, videos, graphics, and promotional text.

- Configuring options and uploading assets.
  Google Play lets you target your application to a worldwide pool of users and devices. By configuring various Google Play settings, you can choose the countries you want to reach, the listing languages you want to use, and the price you want to charge in each country. You can also configure listing details such as the application type, category, and content rating. When you are done configuring options you can upload your promotional materials and your application as a draft (unpublished) application.

- Publishing the release version of your application.
  If you are satisfied that your publishing settings are correctly configured and your uploaded application is ready to be released to the public, you can simply click Publish in the developer console and within minutes your application will be live and available for download around the world.

5.11 LEGAL REQUIREMENTS

License to Use Content. Following payment of the applicable fees for Content, you will have the non-exclusive right, for the period selected by you in the case of a purchase for a rental period, and in other cases for as long as Google and the applicable copyright holder have rights to provide you that Content, to download or stream, in each case, solely as expressly permitted by Google via the Google Play user interface and subject to the restrictions set out in the Terms and associated policies, copies of the applicable Content to your Devices, and to view, use, and display the Content on your Devices or as otherwise authorized by Google as part of the Service for your personal, non-commercial use only. All rights, title and interest in Google Play and Content not expressly granted to you in the Terms are reserved by Google and its licensors.

Violation of License Terms. If you violate any of the terms and conditions of the Terms, your rights under this license will immediately terminate and Google may terminate your access to Google Play, the Content and/or your Google account without refund to you. No Public Performance. You must not display (in part or in whole) the Content as part of an any public performance or display even if no fee is charged (except where such use would not constitute a copyright infringement or violate any other applicable right). Use of a tool or feature provided as an authorized part of Google Play (for example, “Social Recommendations”, as defined in the Music terms below) is permitted, provided that as you use the tool or feature as specifically permitted and only in the exact manner specified and enabled by Google.

Sale, Distribution or Assignment to Third Parties. You may not sell, rent, lease, redistribute, broadcast, transmit, communicate, modify, sublicense or transfer or assign any Content or your rights to Content to any third party without authorization, including with regard to any downloads of Content that you may obtain through Google Play. Use of any tool or feature provided as an authorized part of Google Play (for example, “Social Recommendations”) shall not violate this provision so long as you use the tool as specifically permitted and only in the exact manner specified and enabled by Google.

Capturing of Streams. You may not use Google Play or any Content in conjunction with any stream-ripping, stream capture or similar software to record or create a copy of any Content that is presented to you in streaming format.

Sharing. You may not use Content as part of any service for sharing, lending or multi-person use, or for the purpose of any other institution, except as specifically permitted and only in the exact manner specified and enabled by Google (for example, through “Social Recommendations”).
Security Features. You may not attempt to, nor assist, authorize or encourage others to circumvent, disable or defeat any of the security features or components, such as digital rights management software or encryption that protect, obfuscate or otherwise restrict access to any Content or Google Play. If you violate any security feature, you may incur civil or criminal liability.

Proprietary Notices. You may not remove any watermarks, labels or other legal or proprietary notices included in any Content, and you may not attempt to modify any Content obtained through Google Play, including any modification for the purpose of disguising or changing any indications of the ownership or source of Content.

All rights of the Smart Door project belongs to Ambient Intelligence Lab at Computer Engineering Department of Sharif University of Technology.

6 DESIGN

6.1 SYSTEM DESIGN

Deriving design requirements and clarifying developers’ objectives are explicated objectives need to be further clarified and elaborated to derive design requirements in Android ecosystem.

6.1.1 DESIGN GOALS

Some questions like “what factors influence or increase the fun in android” has been tried to be answered in this system. We challenged ourselves to create a visual language for our users that synthesizes the classic principles of good design with the innovation and possibility of technology and science. This is material design.

6.2 CURRENT SOFTWARE ARCHITECTURE

Figure 22 represents our current software architecture.

6.3 PROPOSED SOFTWARE ARCHITECTURE

6.3.1 OVERVIEW

Our first proposed system architecture is shown in the Figure 23.

The list below expresses the problems with the proposed architecture that are solved in the current version:

- Objects in the architecture proposal should be focusing on main components (e.g. door, cloud, NFC, …)
- Pictures used in the architecture proposal should contain structures to ease the readability, Door should be the main component (showing characteristics of the system)
- System architecture should contain System image, UI and other information we would like to inform to users (User Centric Design, Mental Models)
- Alternatives: Ensuring satisfaction for reviewer
- To allocate numbers to work/data flow
Figure 22. Current System Architecture
Texts and components should be somehow differentiated in first look (e.g. by color)

System architecture specification for technical systems

The very first aspect of a paper is to be replicable, and the key element to make a research replicable is to indicate main components

System architecture should express technologies used in the system

The door itself, the interaction start and the flow should be highlighted in system architecture

Diverse kind of messages (human to human, application to application, application to human…)

To Differentiate the concept of research and the implementation of the concept

6.3.2 SUBSYSTEM DECOMPOSITION

Figure 24 shows all subsystems of the smart door project.

6.3.3 HARDWARE/SOFTWARE MAPPING

QRCode or NFC tags should be available in doors in addition to the existence of the cloud server and android smart phones to run the project’s Android mobile application for interconnecting the client side as an interface to the system.
6.3.4 ACCESS CONTROL AND SECURITY

The owner of the system (the professor in the academic environment use) will create diverse access levels and grant them to the clients (students). Each client might have multi-level access, but having the same access for more than one user will not deny the privacy of the system since users will only access their own archived information. Customers will use unique identifiers (e-mails in the current version) to create smart door accounts and therefore can be tracked by the system owner to prevent unwanted activities.

6.4 USER INTERFACE

Since we were aiming to design an application theme suitable for both student and professor use, we had to consider numerous details and behavioral models user might have to the user interface. 16 application themes were designed. Figure 25 represents our proposed user interface designs.

We asked both our professor and our colleges to give us feedbacks with the designs and finally under supervision of our professor Dr. Ali Asghar Nazari Shirehjini final them of the user interface was chosen. Figure 26 shows the final theme of the application.
To show how the proposed framework can be used, we describe the design and shared the process for two groups of students: AMI lab members and HCI course students. On the next step we will release the application for other professors at the Computer Engineering Department of Sharif University of Technology and then for other departments at this university.

### 7.1 FEATURES TO BE TESTED

We have aimed to test the features that are implemented in the current version of the project. These features include submitting presence, replying to professor’s messages, logging in to smart door accounts, reading doors’ identity via NFC or QR code tags and to fetch the doors’ messages from the server.
7.2 PASS/FAIL CRITERIA

The list below expresses expectations from the user’s point of view that should be met by the system:

1. Users must be able to read the door's identity from QR code or NFC tags in less than 4 seconds.

2. The door’s information must be fetched from the server in less than 10 seconds including last 20 messages exchanged between the student and the professor.

3. Users must be able to reply to professors' message or submit their presence within seconds.

4. All potential errors (e.g. being unable to read tags or not being connected to the internet) should be defined in the system and be responded with proper messages to the end user.

5. The client side of the system (Android mobile application) must never crash (even in the absence of the server side).
7.3 APPROACH

As user-centered designers and developers we seek to use augmented objects to solve real user needs. In this way, the definition of the problem is the first stage we need to address. When a problem is identified, it should be analyzed from various perspectives in order to solve it. Sometimes the problem solution requires automation using computer systems and related technologies. Smart door is aimed to be used in academic environments. Internet connection, Android smartphones and QRCode or NFC tags are also required for using this system.

7.4 TEST CASES

Smart door’s mobile application is released to the ambient intelligence lab members and to HCI course students in order to evaluate the system’s functionality. Figure 27 indicates the test case number one, Mrs. Mostafazadeh, graduate student and Figure 28 indicates second test case, Mr. Halvachi a PhD student. Both test cases are also members of the ambient intelligence laboratory.

![Figure 27. Test Case 1](image1)

![Figure 28. Test Case 2](image2)

8 PROJECT ISSUES

8.1 OPEN ISSUES

The table below indicates our current open issues with the system.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart phone’s capability</td>
<td>There are some concern with smart phones that do not have the ability to read NFC tags and have low quality cameras. Since there are only two option of reading NFC or QRCode tags to access door’s identity, users with addressed problems would not be able</td>
</tr>
</tbody>
</table>
to use the system properly.

<table>
<thead>
<tr>
<th>Professor's interface to interact with the system</th>
<th>At the current stage there is only Parse’s web interface option available for the professors to interact with the system (e.g. adding or modifying students, uploading new messages and etc.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of request</td>
<td>Current version of Parse cloud server only handles 30 request per seconds for free. With increase in the number of the users, other services of this cloud server must be purchased and this issue can cost money.</td>
</tr>
<tr>
<td>Email confirm for sign-up</td>
<td>The current version of the system doesn’t provide the email confirmation for sign up process which can bring potential fraud usage for the system.</td>
</tr>
<tr>
<td>Non-android based interfaces</td>
<td>At the current stage, the only valid interface for students to interact and benefit the system’s services is to have an Android based smart phone with 4.0.3 or higher version of operating system.</td>
</tr>
<tr>
<td>ID sharif</td>
<td>Our current system is running in Department of Computer Engineering at Sharif University of Technology. Users at this university must use a vpn named “ID sharif” that can be challenging for the internet connection and server respond time.</td>
</tr>
<tr>
<td>System’s evaluation and effect</td>
<td>The evaluation process of the system to judge its effect on the student-professor interaction and improving the subjective satisfaction is an open issue now.</td>
</tr>
</tbody>
</table>

8.2 POTENTIAL USER PROBLEMS

As mentioned before the minimum version of Android operation system to support the current version of the application is 4.0.3. Therefore users who do not meet the minimum requirement in this section must update their operating system in order to be able to use the system. There is no current iOS version of an end-user’s application to interact with the system. Therefore students and professors who don’t have an Android based smart phone will not be able to use the current version of the system. Existence of an android based smart phone is crucial for the users to use the current version of the system. Nevertheless we're aiming to create a web based portal for professor to provide them with the ability to use the system without needs of having a smart phone. An iOS client side application is also intended to be implemented for prospective use of the system.

8.3 COSTS

The current version of the system uses Parse cloud server to process user’s request and to respond to them. We can handle 30 requests per second for free, but with the increase in number of user requests, 100$ per each 10 requests must be provided. In addition the current version of the system does not provide file sharing. Parse cloud server is able to provide this feature with 20 GB storage for free and 0.03$ fee for each 1GB extra data storage space.
8.4 WAITING ROOM

The table below indicates all the features that are defined but will not be implemented for the next version of the system.

*Table 8. Prospective Features*

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opening the door.</td>
<td>The door owner being able to open the door by his smart door’s application if the access is granted.</td>
</tr>
<tr>
<td>Context awareness.</td>
<td>The system must be able to confirm the users’ identity by their unique attributes (e.g. finger print, eye print, weight and height…)</td>
</tr>
<tr>
<td>Embed processing ability to the door</td>
<td>The door itself must be able to process users’ requests and be able to respond to them.</td>
</tr>
<tr>
<td>Local network</td>
<td>Doors must provide local networks in order to be able to push notification to both students and professors’ client side interface.</td>
</tr>
<tr>
<td>Network of doors</td>
<td>Each unity of doors in a specific environment must be aware of other every door entity in its surroundings to conduct a network of doors.</td>
</tr>
</tbody>
</table>

We also intend to improve our login subsystem. The figure below shows the prospective login process decomposition.
Ubiquitous computing, pervasive computing, Internet Protocol, sensing technologies, communication technologies, and embedded devices are merged together in order to form a system where the real and digital worlds meet and are continuously in symbiotic interaction. The smart object is the building block of IoT vision. By putting intelligence into everyday objects, they are turned into smart objects able not only to collect information from the environment and interact/control the physical world, but also to be interconnected, to each other, through Internet to exchange data and information. The expected huge number of interconnected devices and the significant amount of available data open new opportunities to create services that will bring tangible benefits to the society, environment, economy and individual citizens.

Now if things such as everyday objects, artifacts, products, etc. could interact with people, they would be able to tell us about the world around them, its history and traditions, and also about the people they got in touch with, their experiences and emotions.
• Internet of Things (IoT):

The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of almost 50 billion objects by 2020.

• Augmented Object:

There are many examples of augmented objects in the literature. Augmented objects should provide intelligence to the ambient where they are located and also they must require a low cognitive effort to be used. Few works have been reported providing guidelines to conceive and design these components. Therefore, developers have to use improvised ad hoc software processes to support the development of augmented objects. In order to help dealing with this situation, this article presents a software process to develop these components. The proposed process was named Augmented Objects Development Process (AODeP) and it is based on the authors’ previous experiences and software engineering best practices. The article also reports two case studies in which AODeP was used to guide the development of augmented objects in specific problems. The obtained results are encouraging.

• Cloud Assisted:

A cloud server is a logical server that is built, hosted and delivered through a cloud computing platform over the Internet. Cloud servers possess and exhibit similar capabilities and functionality to a typical server but are accessed remotely from a cloud service provider. A cloud server may also be called a virtual server or virtual private sever. A cloud server is primarily an Infrastructure as a Service (IaaS) based cloud service model. There are two types of cloud server: logical and physical. A cloud server is considered to be logical when it is delivered through server virtualization. In this delivery model, the physical server is logically distributed into two or more logical servers, each of which has a separate OS, user interface and apps, although they share physical components from the underlying physical server. Whereas the physical cloud server is also accessed through the Internet remotely, it isn’t shared or distributed. This is commonly known as a dedicated cloud server.

• Cyber-Physical Systems (CPS):

Due to the huge advancements in the fields of electronics and the deployments of wireless communication systems, mobile devices and ubiquitous services (providing
anytime-anywhere connectivity to the users) spread rapidly over the past decade. Today, however, the role played by devices is no longer limited to connect users to the Internet, but it has been expanding becoming an opportunity to interlink the physical world with the cyber world, leading to the emergence of Cyber-Physical Systems (CPS). The notion of CPS refers to a next generation of embedded ICT systems where computation and networking are integrated with physical processes and they control and manage their dynamics and make them more efficient, reliable, and adaptable and secure.

- **NFC tag**

  NFC (near field communication) is a wireless technology which allows for the transfer of data such as text or numbers between two NFC enabled devices. NFC tags, for example stickers or wristbands, contain small microchips with little aerials which can store a small amount of information for transfer to another NFC device, such as a mobile phone. There's a whole set of different data types you can store on an NFC tag. The actual amount of data varies depending on the type of NFC tag used - different tags have different memory capacities. For example, you may choose to store a URL (web address) or a telephone number. A standard Ultralight NFC tag can store a URL of around 41 characters, whereas the newer NTAG203 NFC tag can store a URL of around 132 characters. Usually, this information is stored in a specific data format (NDEF - NFC data exchange format) so that it can be reliably read by most devices and mobile phones.

- **Android**

  Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touchscreen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics. As of 2015, Android has the largest installed base of all operating systems.

- **Google Play**

  Google Play, originally the Google Play Store or Android Market, is a digital distribution platform operated by Google. It serves as the official app store for the Android operating system, allowing users to browse and download applications developed with the Android SDK and published through Google. Google Play also serves as a digital media store, offering music, magazines, books, movies, and television programs. It previously offered Google hardware devices for purchase until the introduction of a separate online hardware retailer, Google Store, on March 11, 2015. Applications are available through Google Play either free of charge or at a cost. They can be downloaded directly to an Android or Google TV device through the Play Store mobile app, or by deploying the application to a device from the Google Play website. Many applications can be targeted to specific users based on a particular hardware attribute of their device, such as a motion sensor (for motion-dependent games) or a front-facing camera (for online video calling).
QR code (abbreviated from Quick Response Code) is the trademark for a type of matrix barcode (or two-dimensional barcode) first designed for the automotive industry in Japan. A barcode is a machine-readable optical label that contains information about the item to which it is attached. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte/binary, and kanji) to efficiently store data; extensions may also be used.

The QR Code system became popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC barcodes. Applications include product tracking, item identification, time tracking, document management, and general marketing.

A QR code consists of black modules (square dots) arranged in a square grid on a white background, which can be read by an imaging device (such as a camera, scanner, etc.) and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data are then extracted from patterns that are present in both horizontal and vertical components of the image.

Parse
Parse can pride the system with powerful cloud data base, versatile push notification services, analytical tracking and more.

Mailgun
The Mailgun API is built on HTTP and is RESTful and it:
- Uses predictable, resource-oriented URLs.
- Uses built-in HTTP capabilities for passing parameters and authentication.
- Responds with standard HTTP response codes to indicate errors.
- Returns JSON.
- Mailgun has published Libraries for various languages. You may use the libraries, or your favorite HTTP/REST library available for your programming language, to make HTTP calls to Mailgun.

Bitbucket
Bitbucket is a web-based hosting service for projects that use either the Mercurial (since launch) or Git (since October 2011) revision control systems. Bitbucket offers both commercial plans and free accounts. It offers free accounts with an unlimited number of private repositories (which can have up to five users in the case of free accounts) as of September 2010, but by inviting three users to join Bitbucket, three more users can be added, for eight users in total. Bitbucket is written in Python using the Django web framework.
1. The Internet of Things vision: Key features, applications and open issues, Eleonora Borgia, Institute of Informatics and Telematics (IIT), Italian National Research Council (CNR), via G. Moruzzi 1, 56124 Pisa, Italy
2. Distributed user interfaces in public spaces using RFID-based panels, Ricardo Tesoriero a,n, Pedro G. Villanueva a, Habib M. Fardoun b, Gabriel Sebastián Rivera, University of Castilla-La Mancha, ISE Research group, Information System Department, Campus Universitario de Albacete, 02071 Albacete, Spain