

**Information System Analysis I**

**Taught by Nariman Mansour**

**Smart Laundry Room**

**TEAM KOL**

**Team members**: Keon Woo Park (1831319)

Olive Mukeshimana (1913911)

D Manh Long Nguyen (1911961)

**Course number**: 420-AS2-AS (sect. 07134)

**Submission date**: November 27, 2019

Table of Contents

[Ⅰ. Introduction 2](#_Toc25753721)

[Ⅱ. P.I.E.C.E.S. 2](#_Toc25753724)

[Ⅲ. Requirements 3](#_Toc25753731)

[Ⅳ. Feasibility Analysis 4](#_Toc25753732)

[[Technical Feasibility] 4](#_Toc25753733)

[[Operational Feasibility] 5](#_Toc25753734)

[[Economical Feasibility] 5](#_Toc25753735)

[[Cultural Feasibility] 6](#_Toc25753736)

[[Scheduling Feasibility] 7](#_Toc25753737)

[Ⅴ. Use case 8](#_Toc25753738)

[Ⅵ. Product Backlog 11](#_Toc25753739)

[[Product backlog] 11](#_Toc25753740)

[[User story details] 13](#_Toc25753741)

[Ⅶ. Process and Data: DFD, ERD, class diagram 15](#_Toc25753742)

[[DFD (Data Flow Diagram)] 15](#_Toc25753743)

[[Entity Relation Diagram (ERD)] 16](#_Toc25753744)

[[Class Diagram] 16](#_Toc25753745)

[Ⅷ. Design 17](#_Toc25753746)

[Ⅸ. Conclusion 19](#_Toc25753747)

X. [Appendix 20](#_Toc25753748)

[[Appendix #1 Cost-benefit Analysis] 20](#_Toc25753749)

[[Appendix #2 Scheduling Feasibility (Gantt method)] 23](#_Toc25753750)

[[Appendix #3 Punishment policy] 24](#_Toc25753751)

[[Appendix #4 Product Backlog] 25](#_Toc25753752)

[[Appendix #5 Trello board] 27](#_Toc25753753)

[[Appendix #6 DFD level 0] 28](#_Toc25753754)

# Ⅰ. Introduction

## [Company Description]

KOL is a new company who focuses on developing a smart-laundry room application. We are currently working on a project to launch the company. Currently, in many apartments, people are spending more efforts than they need to in order to do their laundry with public machines. Time isn’t being utilized effectively, and there is little to no security for their laundry. This decreases their quality of life and we saw the problems in the current system of using laundry machines in apartments. Time scheduling as well as the security for laundry machines should be secured and automated. Therefore, we try to create a program as a solution to the current problem so that people can set up their laundry time effectively and conveniently.

## [Scope]

KOL is basically a mobile application, which can be run on every mobile environment supporting internet connection. It is also available on web browsers. The project will serve residents as a helping hand in the apartment. The application, connected to the public laundry room in their apartments, enables residents to book their time to laundry and to keep their laundry in secure. The application allows users to create an account, see the availability of the machines, book and pay their laundry, and customize notification settings.

# Ⅱ. P.I.E.C.E.S.

## [Performance]

In current public laundry rooms, residents have to physically check if laundry machine is free and to stay around to check on their clothes. Also, they need to prepare coins to pay the laundry.

## [Information and Data]

Residents don’t have clear information on if there is machine available, what time the laundry is done and if their laundry is safe. Also, in the current system, residents cannot report back to other residents that they are done with the machine and that the machine is free.

Machines might be free but none of the residents know unless they actually go down to the laundry room, which causes inefficiency.

A laundry machine can be open by other people, which harms the security.

## [Economics]

Currently, tangible costs depend on the tagged price of each room. Therefore, it is inappropriate to say the cost is high. However, intangible costs are too high because residents must put additional efforts other than just money in order to do their laundry. Hence, although the product would not decrease residents’ spending on money, it will hugely reduce the intangible costs, which leads to huge profits.

## [Control and Security]

Currently, there are little to zero security when it comes to apartment laundry machines. People can easily steal others’ clothes.

## [Efficiency]

The current system is inefficient. People waste time when they do not take the finished laundry in time. Also, in many cases, even though there are laundry machines available, no one knows it until they physically are there, which also leads to the time-inefficiency. In addition, people have to stay around the machines while their laundry is in process if they make sure whether someone steals their laundry.

## [Service]

Users cannot be sure if their laundry is done unless they stay near it. Also, the current system is inflexible to new or exceptional situations such as stealing.

# Ⅲ. Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Non-functional Requirements | | | |
| operational | performance | security | Cultural & political |
| Can be run on both mobile devices and computer   * can be run on all web browsers | Users can leave feedbacks | Laundry machines will be opened only by code which will be given to the user during booking process | User can choose the colour theme of application. |
| User can set the notification schedule according to their preference | Update machine information every 10 seconds | Only system managers can see the personal records of users. | Can be run in French or English. |
| Can be run on any environment that provides internet connection | Users will get the responses to their requests within a period of 15 seconds | Users can see their history of using the system | Shall not use icons that could be considered offensive in our market countries |
|  |  | Only authenticated user can use the system | Support Canadian dollar currency |

|  |  |
| --- | --- |
| Functional requirements | |
| Process | Information |
| Allow users to book the time to laundry   * also to cancel the booking * also to pay when booking | Include real-time information on the status of each laundry machine (to be displayed to the user) |
| Notify the user when laundry is done | Include real-time tracking on the booked machine |
| Allow users to create an identification. | Collect user information on the history of using the machine according to reservation / report system |
| Generate a code to unlock the machine (connected to each lock in each machine)   * Provide authenticated users with a code to unlock the machine |  |
| Warn the user who does not take one’s laundry |  |

# Ⅳ. Feasibility Analysis

## [Technical Feasibility]

The application will be developed in Android, iOS and Windows Phone, and also for web-environment.

The technologies we are going to use are familiar. The smart lock system is quite popular now with the advent of IoT, and a laundry machine has nothing new. However, the combination of using a smart lock and a laundry machine might be new.

Programming languages and platforms to be used are easily accessible.

Technologies needed:

Front End:

- C#

- C++

- HTML5

- CSS3

- JavaScript

- JQuery

Back End:

- MongoDB

- SQL / Oracle SQL (Backup for Mongo)

Physical/IRL:

- Smart Lock

## [Operational Feasibility]

For the user-friendly environment, the system allows users to create their accounts which include their information on addresses, names, and preferred laundry options. The information will have to be connected to the corresponding laundry room.

The interface of the application will be easily readable and will not require technical expertise to be accessed.

The system will have security environment for payments, fast processing and response time for user requests.

The application will be able to run in any mobile platforms, including Android, iOS, and Windows.

The system tracks real-time data on the machine uses and presents it automatically.

## [Economical Feasibility]

The application development would not cost much compared to other applications on the market. As it will be developed using existing system and hardware, the development time will not be long, which leads to reduction in the developmental cost. Also, it wouldn’t take much cost in testing and implementing because most of users will be already familiar with this kind of technique.

Operational cost will also not be high because the system will be automated. There does not need to be employment on a constant alert.

Detailed cost and benefit of the system is following:

Benefits

- Good will: resolve inefficiency and inconvenience of using public laundry rooms

- Reduction in resident complaints

- Increase in brand recognition

- Advertisement

- Interactive user experience

- Providing accurate information on availability of machines

Developmental Costs

- Hardware and software costs

- Development team salary

- Database building

- Testing and training

- Registration

Operational Costs

- Administrative costs

- Maintenance

- Software upgrade

- Hardware repair

- Communication charge

- Yearly registration fee

The system will cost $14,100 as developmental cost. Total operational costs will be $5,700 per year. And the total benefits will be $10,500 per year. The economical feasibility analysis of nine years shows NPV of $14,825.13, ROI of 44%, and payback period of 2.27 years. Also, the break-even point of the system will be 35 months (2 years and 11 months).

(View Appendix #1 for more details on the Economical feasibility analysis)

## [Cultural Feasibility]

Considering the application will be first implemented in Montreal, it will support two languages for the first version, French and English, so that there will not be any language barrier. Also, the interface of the application will be very intuitive, which will also reduce the language barrier. Once it got big enough to expand to other cities and countries, more currency and language will be added.

All the transactions will be paid in Canadian Dollars.

## [Scheduling Feasibility]

The total development period will be 12 weeks. View Appendix #2 for more details on the scheduling feasibility (using GANTT method).

As a result of the feasibility study and all the preliminary investigation, we know that the current system is not only inefficient, but it is also unsafe. Although there is an existing technology that can reduce the inefficiency and improve the security, it is not being applied. Therefore, we suggest an upgraded system, which is the Smart Laundry Room. In the smart laundry room, all the process of laundry, except bringing and taking out the laundry, is performed online. This makes is possible for the users to check the availability of machines and progress of their laundry. Therefore, this system reduces the inefficiency of the current system by eliminating the case where a user brings his/her laundry but no machine is available. In addition, this system ensures the security of laundry with its lock system. Locks installed to laundry machines will be opened only by a legitimated user, who reserves the machine. In conclusion, our system contributes to improve the quality of life by making laundry, which is essential in life, easier.

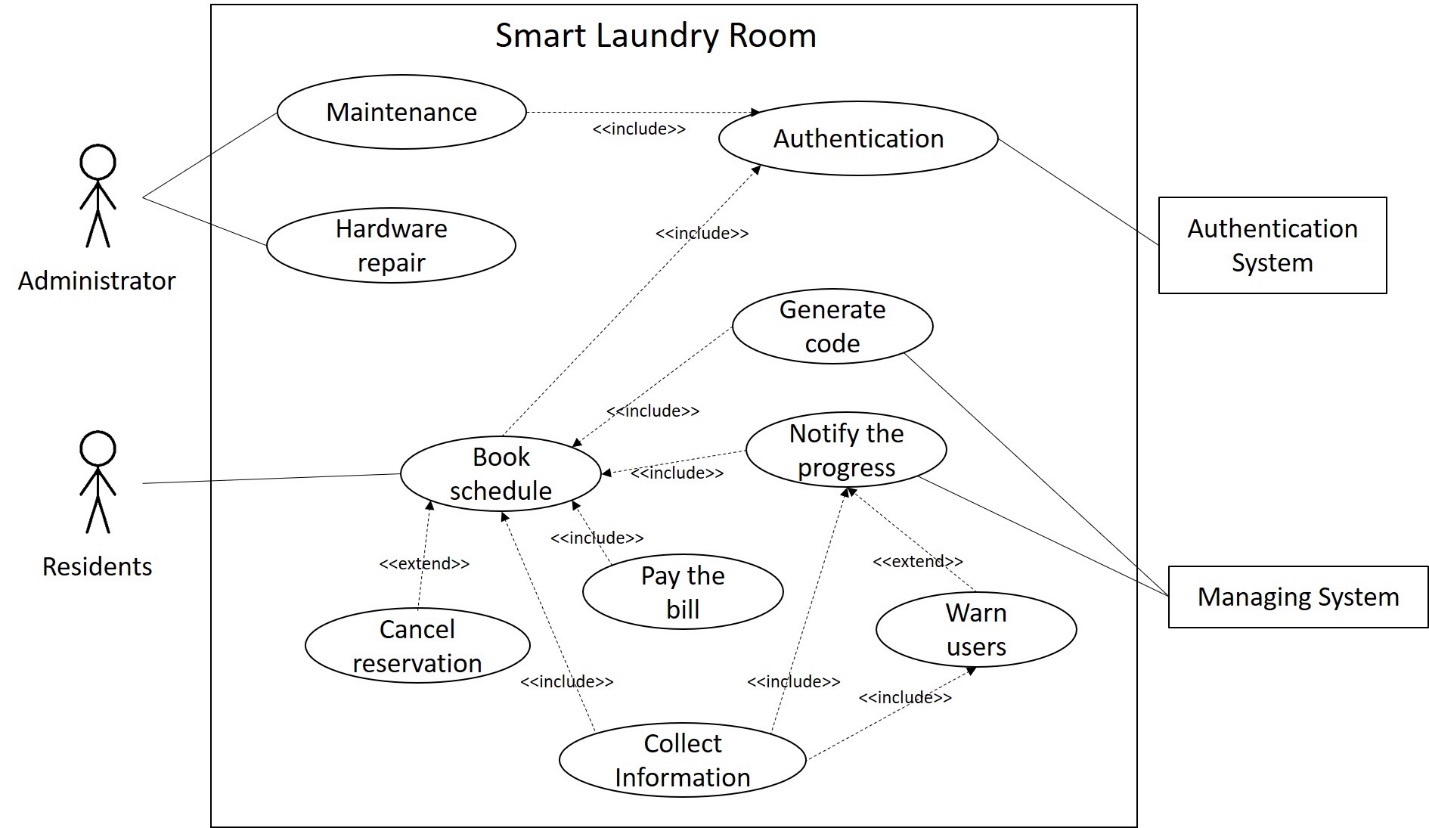
# Ⅴ. Use case

Three detailed use cases are included for critical functionalities: reserve a laundry, register and notify progress.

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name: reserve a laundry | | Id: 1 | |
| Short description: this feature allows a user to reserve time to laundry | | | |
| Trigger: user activates the reservation form on the application | | | |
| Type: External | | | |
| Major Inputs | | Major Outputs | |
| Description | Source | Description | destination |
| 1. reservation form  2. user information  3. availability | 1. user  2. user DB  3. system DB | 1. Successful notification message with code to open the machine  2. error message  3. revised user information  (tracking the usage)  4. revised availability  (time, machines) | 1. user  2. user  3. user DB  4. system DB |
| Major steps performed | | Information for steps: | |
| 1. A user requests (fills out) the booking form containing information about time to laundry, amounts of laundry, heat setting… | | [I.1] Reservation form | |
| 2. The system checks the user information and availability and display the time slot | | [I.2] User information, machine availability  [O.2] Filled form | |
| 3. If all the information from the user is valid, a successful notification message will be displayed with the code to open the machine. | | [I.3] Valid form  [O.3] Successful notification message | |
| 4. If the information is wrong or invalid, it gets rejected and an error message is displayed | | [I.4] Invalid form  [O.4] Error Message | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name: register | | Id: 2 | |
| Short description: this feature allows a user to create an account for the application | | | |
| Trigger: User who doesn’t have an account clicks the button register | | | |
| Type: External | | | |
| Major Inputs | | Major Outputs | |
| Description | Source | Description | destination |
| 1. Registration form  2. User information | 1. user  2. user | 1. Successful notification message  2. Error message  3. User information | 1. user  2. user  3. user DB  4. system DB |
| Major steps performed | | Information for steps: | |
| 1. A user fills out an online registration form that contains his/her personal information (username, password, full name, address, e-mail address…) | | [I.1] Registration form | |
| 2. Validate the filled registration form | | [I.2] User information  [O.2] Validated information | |
| 3. If all the information from the user is valid (username should be distinct, address should be one of the supporting building, etc.), a successful notification message will be displayed and a confirmation e-mail will be sent to the entered e-mail. | | [I.3] Valid form  [O.3] Successful notification message,  Confirmation letter through e-mail | |
| 4. If the information is wrong or invalid, it gets rejected and an error message is displayed | | [I.4] Invalid form  [O.4] Error Message | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name: notify progress | | Id: 3 | |
| Short description: System notifies each user about the progress of the laundry and give alarm according to user setting | | | |
| Trigger: user locks the lock after starting the laundry | | | |
| Type: External | | | |
| Major Inputs | | Major Outputs | |
| Description | Source | Description | destination |
| 1. Lock  2. User setting | 1. system  2. userDB | 1. Laundry progress display  2. Alarm | 1. user  2. user |
| Major steps performed | | Information for steps: | |
| 1. When the lock is locked, the application will display the time bar that represents the progress of the laundry. | | [I.1] Locked  [O.1] Progress bar | |
| 2. According to user setting (e.g. 10 minutes before laundry is done, 5 minutes …), application will cause an alarm to the user | | [I.2] User setting  [O.2] Alarm | |
| 3. When the laundry is done, user will be notified (every 5 minutes after) | | [O.2] Notification | |



The system works based on the use cases. A resident, as a main actor, uses the system mainly to book his/her laundry on his/her preference. To use the system, a user must verify himself. Once a user fills the reservation form and pays the bill, the system collects the information and generates a progression schedule for the laundry in order to notify the progress. Also, if a user does not open the lock to take out the laundry in certain amount of time, the system generates a warning message according to the policy. If a user wants to cancel the reservation, he/she can get the money back according to the policy (view appendix #3 for the details). In all the processes, the system collects information on what is going on to update data.

# Ⅵ. Product Backlog

Here, we suggest the estimated and prioritized product backlog with user stories. User story point is calculated with assumption that we can perform 9 points per sprint. One sprint will be two weeks and as the expected total development time is 12 weeks (refer to the appendix #3), there will be six sprints. The sum of all the user story points is 53, and they will be distributed and developed according to the priority (View appendix #4 for the written form of the product backlog).

## [Product backlog]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PBI | Reasoning/arguments | priority | Points |
| mandatory | Reservation | User can view the time availability up to three days from the moment, can reserve his/her laundry, and cancel it | 1 | 13 |
| mandatory | User Identification | User can enter their username and password to access the system. If not registered, they can register. | 2 | 7 |
| mandatory | Notification | User will be notified the progress of laundry. User can set the notification option | 3 | 7 |
| mandatory | Pay | User can pay the laundry with debit / credit card. User can preset their payment option | 4 | 7 |
| mandatory | Lock | System can generate a code for the eligible user to unlock the laundry machine  User can get the code when they finish booking and use it to open the machine | 5 | 8 |
| mandatory | Select lanugage | If a player reveals a mine, it's a loss, if a player reveals all squares that do not contain mines he/she wins | 6 | 4 |
| optional | Check personal history usage | User can check their history of machine use | 7 | 3 |
| optional | Change color theme | User can change the color theme of the application according to the preference | 8 | 4 |

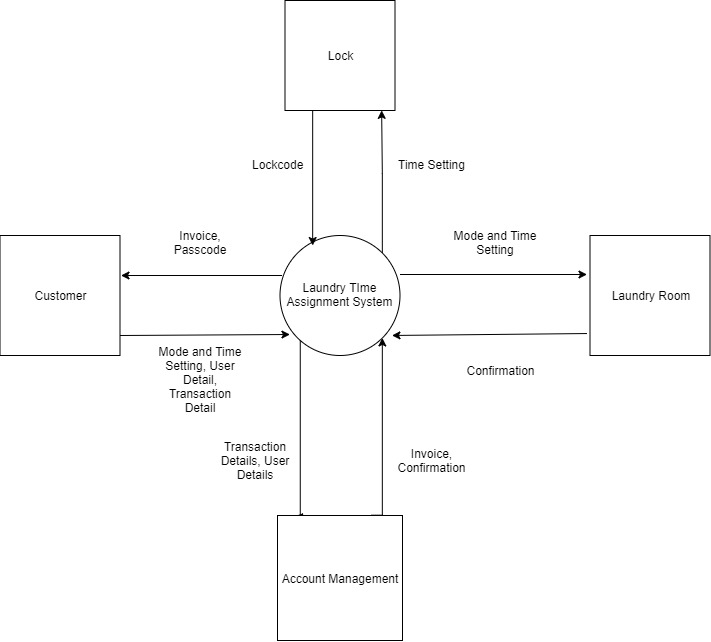
## [User story details]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PBI | As a | I want to | So that | Test criteria | Point |
| Reservation | User | View the available time slot for up to three days | I can plan my laundry | Can show the correct time slot corresponding to user address | 5 |
| User | Reserve time for laundry | I can do my laundry in my preferred time | Can see the time slot which indicates availability Can choose the empty slot (which means available) Cannot choose a already reserved slot | 5 |
| User | Cancel my reservation |  | Can get rid of the reservation Can compare the reserved time and the time when cancellation happens and penalize the user if needed according to the rule | 3 |
| User Identification | Tenant | Create an account | I can be registered as a user from a specific building | Can select registration button and fill the form and submit it Can enter user information including username, password, and apartment code | 3 |
| User | Login to the system | I can access to all the functions of the application | Can enter username and password and login | 3 |
| User | Logout from the system | I can deactivate my account when I don't use it |  | 1 |
| Notification | User | Customize the notification setting | I can take my laundry in time | Can select the time for notification message to be sent (e.g. 5 minutes before the laundry is done) | 2 |
| User | Check the progress / current state of my laundry | I can take my laundry in time | Can see a timeline indicating laundry progress | 5 |
| Pay | User | Preset my payment option | I can automate my payment for laundry | Can make a credit or debit card registered | 3 |
| User | Pay by credit card |  | Can pay by either mastercard or visa | 2 |
| User | Pay by debit card |  |  | 2 |
| Lock / Unlock | User | Receive the code for the laundry machine when my allocated time slot started | I can use the code to open the machine if needed | System can generate the code if the booking is valid The code has to match the lock on the laundry machine | 5 |
| User | Open the laundry machine if needed | I can add or take out clothes I put in the machine | The laundry machine can be opened with the code | 3 |
| Language Selection | User | Use the application in English |  | Can be run in English | 2 |
| User | Use the application in French |  | Can be run in French | 2 |

The user story point for reservation is 13, which is over the estimated capability of a sprint (9). Therefore, reservation will be divided into two parts: View the available time slot for up to three days and Reserve time for laundry; Cancel my reservation. For the example of the development in progress according to the product backlog, view appendix #5.

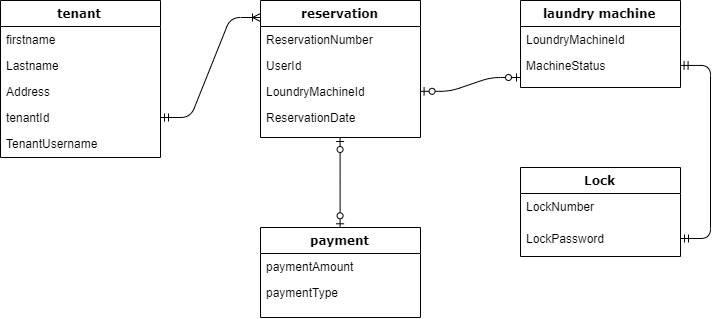
# Ⅶ. Process and Data: DFD, ERD, class diagram

## [DFD (Data Flow Diagram)]



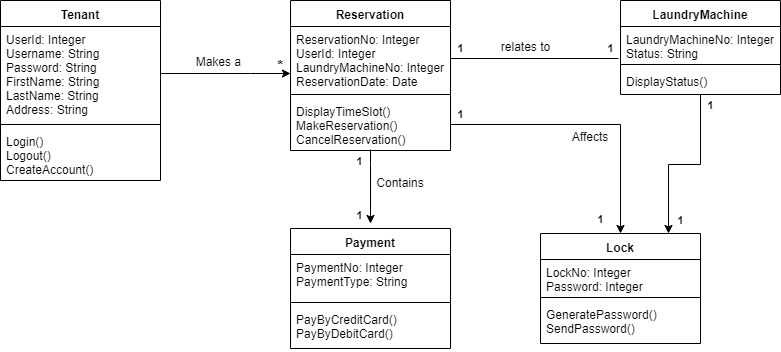
First, customer fills in the information needed to do laundry, which is the mode and time settings, the user details, and the transaction details like credit card number and billing address. Then, the laundry time assignment takes that information and sends it to 3 different sources - lock, laundry room and account management. The system then sends the time setting to the lock system in order to receive back the lock code and the lock duration for the laundry load. In the same time, the laundry time assignment system also sends the mode and time setting to the laundry room so that the machines could be set up before the tenant arrives. The laundry room would send back to the system a confirmation if everything is set up correctly. After the confirmation from the laundry room, the system sends the transaction details as well as the user details to Account Management. Here, Account Management charges the customer and sends back an invoice and a confirmation. At the end, the system sends back the invoice and the passcode to the customer (View Appendix #6 for level 0 DFD).

## [Entity Relationship Diagram (ERD)]



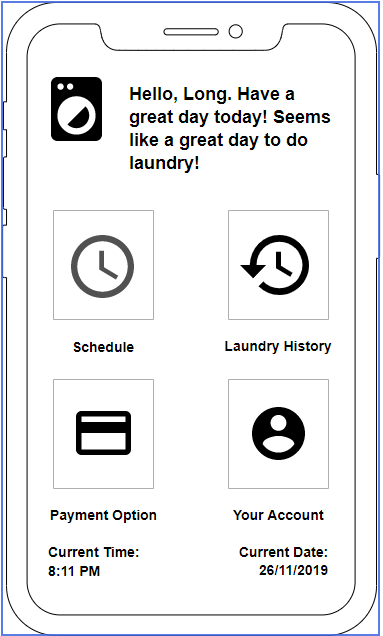
The Entity Relationship Diagram shows how data is organized as the system works. We have five entities and all the events start from a tenant. When a tenant reserves a laundry machine (can be one to many) with valid setting, data is entered in all the other tables.

## [Class Diagram]

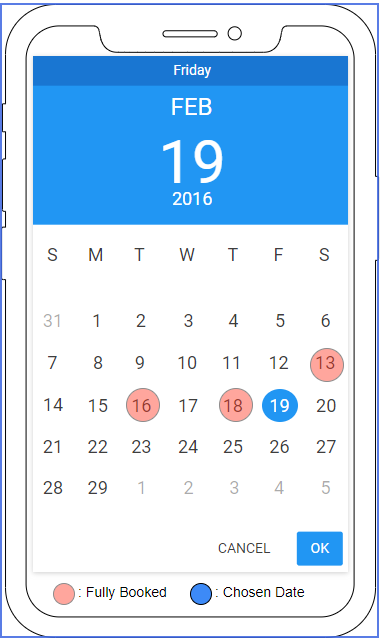
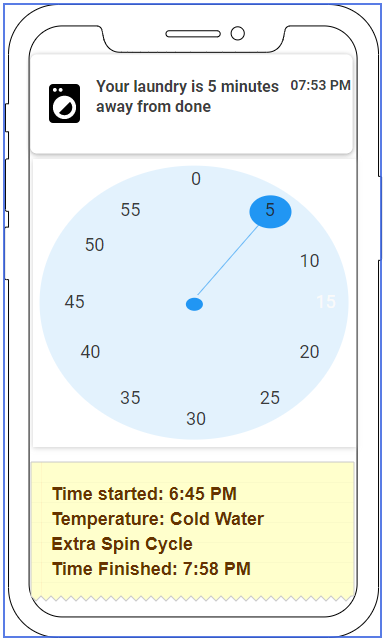


This class diagram also has five entities. A tenant can initiate one or many instances of reservation class. Then, each reservation instance initiates an instance of class laundryMachine, payment and lock. Therefore, this shows a simple process of how the system works and how data is created, updated and moved when an event is initiated.

# Ⅷ. Design

Here are some screens. They give the idea how the application actually works.

This is main screen. If a user logs in to the system, he/she will see this screen. There are just four options: Schedule, laundry history, payment option, and your account. In schedule tab, the user can view the timeslot (daily or hourly) and reserves the laundry. In the second tab, he can check the history of his using the system. In the third tab, he can pre-set his payment option so that he can easily pay the laundry when he reserves. Finally, in your account tab, he can change the account and settings such as notification.

The first screen is displayed when a user wants to look at the daily schedule. If he selects a day to laundry and click OK, the hourly schedule will be displayed. If he chooses an invalid date, which is circled red, an error message will be displayed.

The second screen is notification. When a laundry is almost done, the application will display a notification message according to the user setting. In this example, the user sets to be notified five minutes before his laundry is done.

# Ⅸ. Conclusion

In conclusion, we define the biggest problem of the current public laundry system as inefficiency and insecurity. To improve the current condition, we suggest an alternative: The Smart Laundry Room. In the smart laundry room, there is no more waiting in order to do the laundry. It also ensures the safety of the laundry. Although this will not reduce tangible costs of using the laundry machine, it will hugely reduce intangible costs.

The major obstacles of our project include the followings: a smart lock should be attached to an existing laundry machine; a laundry machine itself cannot be converted; we have to contact apartment owners (or companies) to start building the application; difficulty of estimating economical feasibility and user story points. These obstacles are mostly from the lack of experience in initiating a project and analyzing it with diverse milestones.

Nevertheless, it is meaningful because we, as programmers, can actually do something in order to improve our surroundings. In addition, learning how the real world works teaches us the way to think as an analyst or a programmer. For example, we can now think of the scene behind technologies we use everyday. Finally, we could re-estimate ourselves and be more confident by overcoming the major obstacles.

Team KOL consists of three members: Keon, Olive, and Long. Works for the whole project are almost equally divided. Some works, such as user stories and use cases, are done altogether. And each member takes responsibility to draw a diagram (ERD, class diagram, and DFD). The final document (this report) and presentation is made by Keon using sources prepared by the three members.

# Appendix

## [Appendix #1 Cost-benefit Analysis]

**Developmental Costs**

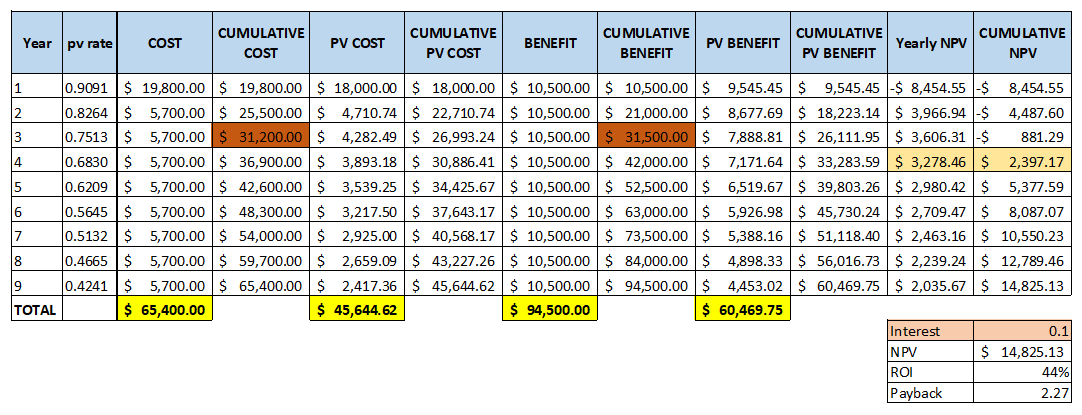
|  |  |
| --- | --- |
| New hardware | $ 5,500.00 |
| Software | $ 1,000.00 |
| Installation | $ 500.00 |
| Design | $ 1,000.00 |
| Development team salary | $ 4,500.00 |
| Database building | $ 1,000.00 |
| Training | $ 500.00 |
| Registration | $ 100.00 |
| Total development cost | $ 14,100.00 |

**Operational Costs**

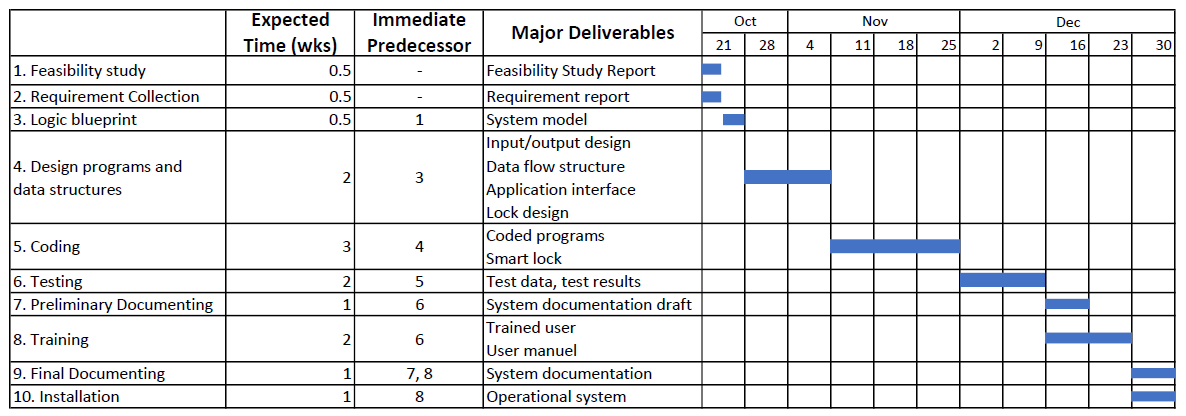
|  |  |
| --- | --- |
| Administrative cost | $ 2,000.00 |
| Maintenance | $ 1,000.00 |
| Software upgrade | $ 600.00 |
| Yearly registration fee (app store) | $ 100.00 |
| Hardware repair | $ 1,000.00 |
| Communication charge | $ 1,000.00 |
| Total operational cost | $ 5,700.00 |

**Benefits**

|  |  |
| --- | --- |
| Good will | $ 4,000.00 |
| Reduction in resident complaints | $ 2,000.00 |
| Increase in brand recognition | $ 1,500.00 |
| Advertisement | $ 3,000.00 |
| Total yearly benefit | $ 10,500.00 |



## [Appendix #2 Scheduling Feasibility (Gantt method)]



## [Appendix #3 Punishment policy]

1. A user can cancel his/her reservation without any charge until two hours before the reserved time. His/her payment will be fully returned.

2. A user must pay 50 cents to cancel his/her reservation if he/she cancels one to two hours before the reserved time. His/her payment will be returned with 50 cents of fine.

3. A user must pay 1 dollar to cancel his/her reservation if he/she cancels in one hour before the reserved time. His/her payment will be returned with 1 dollar of fine.

4. A user must take responsibility for the cancellation 30 minutes before the reserved time. His/her payment will not be returned.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Time before the laundry | **~ 2 hours** | **1 ~ 2 hours** | **30 ~ 60 minutes** | **~ 30 minutes** |
| Fine | **0** | **50 cents** | **1 dollar** | **Full charge** |

## [Appendix #4 Product Backlog]

Product Backlog:

* User stories (Green): What the end user wants from the application and what the programmers need to implement.
* Technical requirements (Orange): Things that need to be built for other stories to be built on top of.
* Code Spikes (Blue): Proofs of concepts or research items that you need to do before you can understand a user story or technical requirement down the road.
* Technical Debt (Purple): Some places you might have to make a decision to short-come a design that you might want to get the product out earlier.
* User stories:

1. User Identification:

* As a tenant, I want to create an account so that I can be registered as a user from a specific building.
  + Test criteria: Can select registration button and fill the form and submit it. Can enter user information including username, password and apartment code.
* As a user, I want to login to the system so that I can access to all the functions of the application.
  + Test criteria: Can enter the username and password.
* As a user, I can logout from the system so that I can deactivate my account when I don’t use it.

1. Reservation:

* As a user, I want to view the available time slot for up to three days so that I can plan my laundry.
  + Test criteria: Can show the correct time slot corresponding to user address.
* As a user, I can reserve time for laundry so that I can do my laundry in my desired time slot.
* Test criteria: I can see the time slot which indicates availability, I can choose the empty slot (which means available), I cannot choose an already reserved slot.
* As a user, I can cancel my reservation.
  + Test criteria: I can get rid of the reservation, I can compare the reserved time and the time when the cancellation happens and penalize the user if needed according to the rule.

1. Payment:

* As a user, I can preset my payment option so that I can automate my payment for laundry.
* As a user, I can pay by credit card.
* As a user, I can pay by debit card.
* As a user, I can pay by either Mastercard or Visa.

1. Notification:

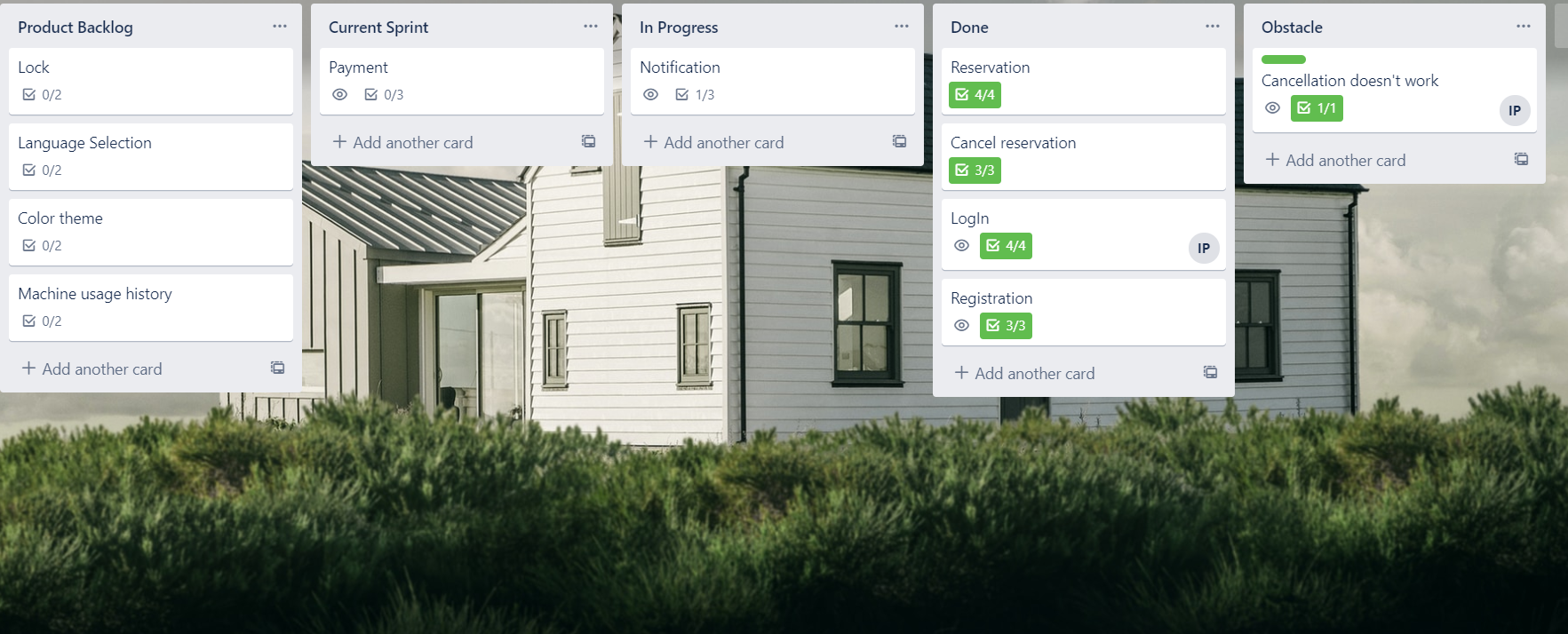
* As a user, I can customize the notification setting so that I can customize the notification setting so that I can take my laundry in time.
  + Test criteria: can select the time for notification message to be sent (e.g. 5 minutes before laundry is finished)
* As a user, I can check the progress / current state of my laundry so that I can take my laundry in time.
  + Test criteria: I can see a timeline indicating laundry progress.

1. Lock / Unlock:

* As a user, I can receive the code for the laundry machine when my allocated time slot started so that I can use the code to open the machine if needed.
  + Test criteria: System can generate the code if the booking is valid, the code has to match the lock on the laundry machine.
* As a user, I can open the laundry machine if needed so that I can add or take out clothes I put in machine.
  + Test criteria: The laundry machine can be opened with the code.
* Technical requirements:
* The lock system
* The application
* Sprints and schedule
* Other artifacts of Scum/Agile.
* Code spike:
* The most optimal language to code the application
* The most convenient way to connect the lock to the system
* The most optimal way to connect the laundry machine to the application
* Technical Debts:
* Skip on making a clean-cut interface
* Skip on making some of the feature in the notification settings

## [Appendix #5 Trello board]

**(**<https://trello.com/b/RAQIOWBS/smart-laundry-room>**)**



## [Appendix #6 DFD level 0]

