



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

Dorm Services Key Rental System (SUZ Services System)

Analysis

Documentation of a project for the purpose of the course BIE-SI1.

Authors: Alejandro Miguel Chirivella Ciruelos, Emily Hannon, Victor Arribas Moreno, Alexander Muryshkin, Andrey Olos, Noam Yehuda, Lawrence Zawahri



Contents

| | | |
|-----------|--|-----------|
| 1. | Project specification..... | 5 |
| 2. | Business Process Model | 6 |
| 2.1 | Dealing With issues | 6 |
| 2.2 | Make and switch reservation | 7 |
| 2.3 | Your reservation..... | 8 |
| 3. | Business Domain Model | 9 |
| 3.1.1 | Complaint..... | 9 |
| 3.1.2 | Reservation | 9 |
| 3.1.3 | Student..... | 9 |
| 3.1.4 | Swap request | 10 |
| 3.1.5 | Washing Machine..... | 10 |
| 3.1.6 | Key | 10 |
| 3.1.7 | Room..... | 10 |
| 3.1.8 | Building | 10 |
| 4. | Requirements..... | 11 |
| 4.1 | Non-functional requirements..... | 12 |
| 4.1.1 | NF01: Web application | 12 |
| 4.1.2 | NF02: Database | 12 |
| 4.1.3 | NF03: UI | 12 |
| 4.1.4 | NF04: Load system of users | 12 |
| 4.2 | Functional Requirements | 13 |
| 4.2.1 | F01.1.2: Keep record of penalty points | 14 |
| 4.2.2 | F01.1.3: Keep record of swap requests | 14 |
| 4.2.3 | F01.1: Identify correct users as students | 14 |
| 4.2.4 | F01.2: Identify correct users as receptionists | 14 |
| 4.2.5 | F01: Keep record of registered users..... | 14 |
| 4.2.6 | F02.1: Add washing machine..... | 14 |
| 4.2.7 | F02.2: Delete washing machine..... | 14 |
| 4.2.8 | F02.3.1: Adjust reservations for suspended machines | 15 |
| 4.2.9 | F02.3.2: Allow to return and remember reservations in case of reopen | 15 |
| 4.2.10 | F02.3: Suspend washing machine | 15 |
| 4.2.11 | F02: Keep record of available machines | 14 |
| 4.2.12 | F03.1: Make reservation | 15 |
| 4.2.13 | F03.2: Cancel reservation..... | 15 |
| 4.2.14 | F03.3: Make swaps between reservations | 15 |
| 4.2.15 | F03: Keep track of reservations..... | 15 |
| 4.2.16 | F04: Keep record of registered users..... | 16 |
| 4.2.17 | F04: Send alerts/notification to users..... | 16 |
| 4.2.18 | F05.1: Penalize student | 16 |



| | | |
|-----------|---|-------------------------------------|
| 4.2.19 | F05: Make a complaint..... | 16 |
| 4.2.20 | F06.1: Generate QR code for reservation | 16 |
| 4.2.21 | F06: Access reservations..... | 16 |
| 4.2.22 | F07: Return key | 16 |
| 5. | Use Case Diagrams | 17 |
| 5.1 | Actors..... | 17 |
| 5.1.1 | Receptionist..... | 18 |
| 5.1.2 | Student..... | 18 |
| 5.1.3 | Student1 | Error! Bookmark not defined. |
| 5.1.4 | Student2 | Error! Bookmark not defined. |
| 5.2 | File a Complaint | 18 |
| 5.2.1 | Fill out complaint form..... | 18 |
| 5.2.2 | Check complaint queue | 19 |
| 5.2.3 | Update complaint with action taken..... | 19 |
| 5.3 | Make/delete a reservation..... | 19 |
| 5.3.1 | Show a list of available time slots | 19 |
| 5.3.2 | Select and reserve/delete reservation..... | 20 |
| 5.4 | Manage Services/Students | 20 |
| 5.4.1 | Add/delete service | 21 |
| 5.4.2 | Show a list of current services | 21 |
| 5.4.3 | Modify service | 21 |
| 5.4.4 | Modify student..... | 21 |
| 5.4.5 | Add/delete students | 21 |
| 5.4.6 | Show a list of current students..... | 22 |
| 5.5 | Return key | 22 |
| 5.5.1 | Access QR and check in | 22 |
| 5.5.2 | Close service..... | 22 |
| 5.6 | Swap request | 23 |
| 5.6.1 | Submit a swap request..... | 23 |
| 5.6.2 | Check alerts/swap requests..... | 23 |
| 5.6.3 | Accept swap request | 23 |
| 5.7 | Use service..... | 24 |
| 5.7.1 | Access QR code | 24 |
| 5.7.2 | Confirm reservation..... | 24 |
| 5.7.3 | Activate usage of service..... | 24 |
| 6. | Domain model..... | 25 |
| 6.1.1 | Complaint..... | 26 |
| 6.1.2 | Issue with machine | 27 |
| 6.1.3 | Issue with student | 27 |
| 6.1.4 | Receptionist..... | 27 |
| 6.1.5 | Reservation | 28 |



| | | |
|--------|----------------------|----|
| 6.1.6 | Student..... | 29 |
| 6.1.7 | Swap request | 29 |
| 6.1.8 | Person..... | 29 |
| 6.1.9 | Washing Machine..... | 29 |
| 6.1.10 | Key..... | 30 |
| 6.1.11 | Room | 30 |
| 6.1.12 | Building..... | 30 |

1. Project specification

The goal of this project is to create a system for key rental to dorm services, specifically designed with laundry key management in mind but applicable to any key rental. The system should assist students in scheduling time to do their laundry at a designated machine (or schedule use of other dorm facilities) and help the receptionist/manager keep track of key rental to the laundry and other dorm facilities.

Information about the students and their various reservations will be stored in the system. Students will be able to view, swap, and modify their reservations and receptionists will be able to view and modify student data as necessary. The system will also allow students to file a complaint with the dorm if the facilities require repair or cleaning or to report incorrect use of facilities by the previous user (i.e. mess/laundry left by previous students, vandalism, etc.). These complaints will be queued for the receptionist to address on their own time. The receptionist will have the ability to penalize students who misuse the facilities.

When renting and returning keys, students will present a unique QR code identifier that will allow the system to keep track of who has rented or returned a key. If a student is unable to return the key at the end of their rental period, the receptionist can penalize them. Students start with three points that may be used each week to rent one hour of laundry time and they may lose these points if the receptionist has a reason to penalize them.

The system will run as a desktop application where the receptionist can manage key rental and student reservations. Students may use either a desktop or mobile viewable web application to create and manage their own reservations and file complaints to the dorm receptionist.

2. Business Process Model

In the domain of the dorm facilities (specifically the laundry facilities), there are three critical business processes – dealing with issues, making and switching reservations, and key rental and return (your reservation).

2.1 Dealing With issues

This section describes the suggested process for dealing with facility issues reported by students. This flow begins when a user with a receptionist account receives any type of complaint from another user. Once the issue is received the issue is analyzed whether it should be considered valid or not. If the decision is made that the complaint is not valid then the receptionist closes the issue and the flow ends. But if the complaint is found to be valid then the issue is accepted. Depending on the issue itself either a student is penalized, or a machine is closed for a given amount of time. These changes are sent to the system where the necessary changes are made for it.

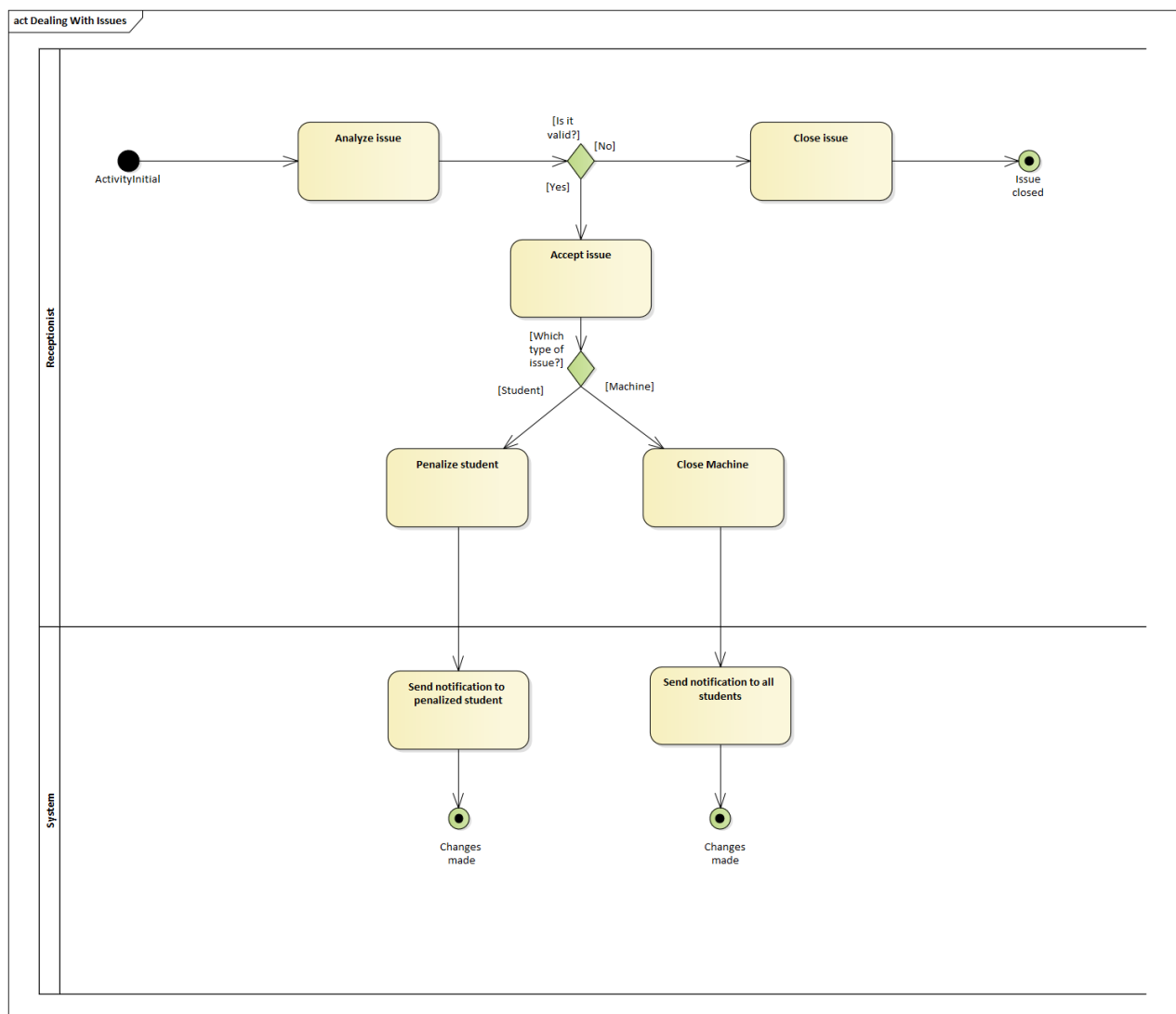


Figure 1 - Dealing with Issues

2.2 Make and switch reservation

This section describes the suggested process to create and modify reservations for the dorm facilities. This flow starts when a user wants to create a reservation. The user begins with selecting a reservation slot from the schedule for a specific machine, if the reservation is already booked then the student can choose to request a swap from the current reservation holder. From there the holder can choose to either reject, which would end the flow there, or accept the request, which would send the changes to the system. But if the requested time is available and the student has enough allowed time left the reservation will be added into the system right away.

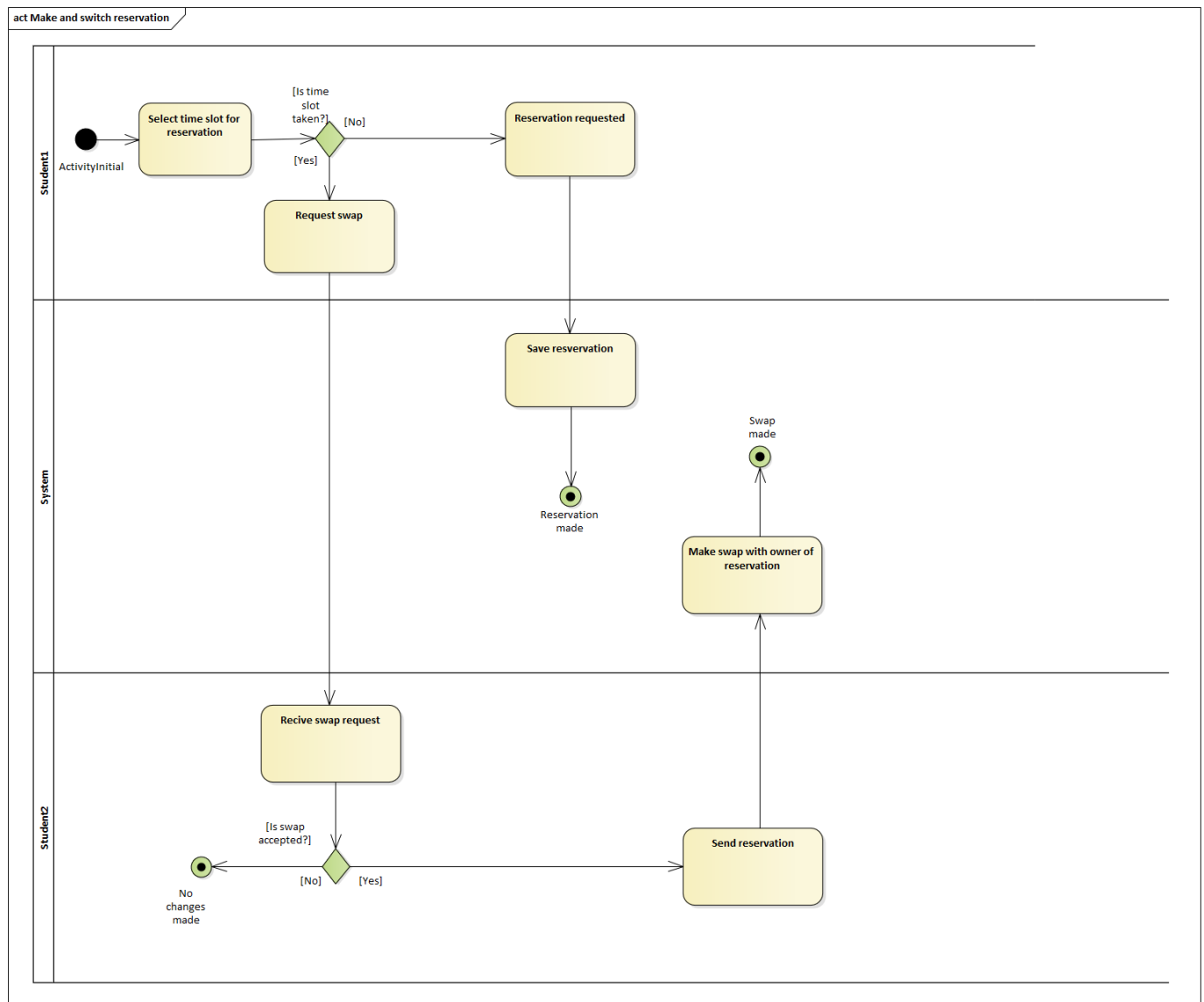


Figure 2 - Make and switch reservation

2.3 Your reservation

This process demonstrates the situation when the student begins their reservation by collecting the key from the receptionist, use of the facilities during the reservation, and return of the key. The flow of this process begins when the student user's reservation time begins. The user will open the reservation which will generate a QR code which they then show to the receptionist, then the receptionist scans the QR to verify that the reservation does indeed belong to the student. If the check reveals that the student is not the reservation holder, then the student is dismissed, and the flow ends there. But if the QR shows that the student does have a reservation then the receptionist needs to look for a key to the correct machine. If the key is not there then the reservation is canceled and the flow ends, if the key is found then it is given to the student and the student can start to use the machine for his allocated amount of time. During this time the student can make complaints either about the machine itself or the condition of the room. After either the time ends or if the machine is no longer needed before that then the key to the machine is returned and the process ends.

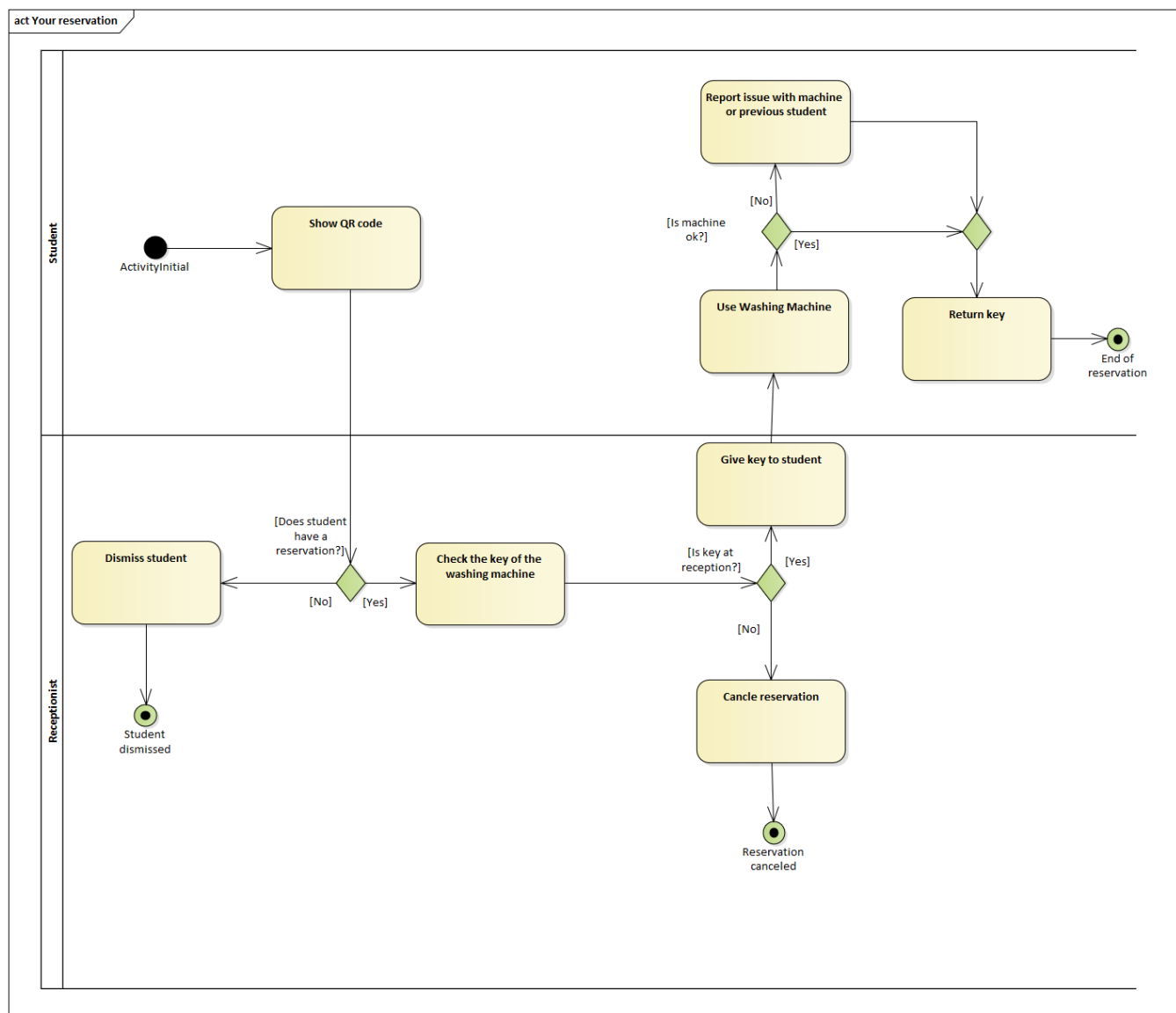


Figure 3 - Your reservation

3. Business Domain Model

The most important domain objects from the dorm facilities domain, interacting in the Business Process Model are described below.

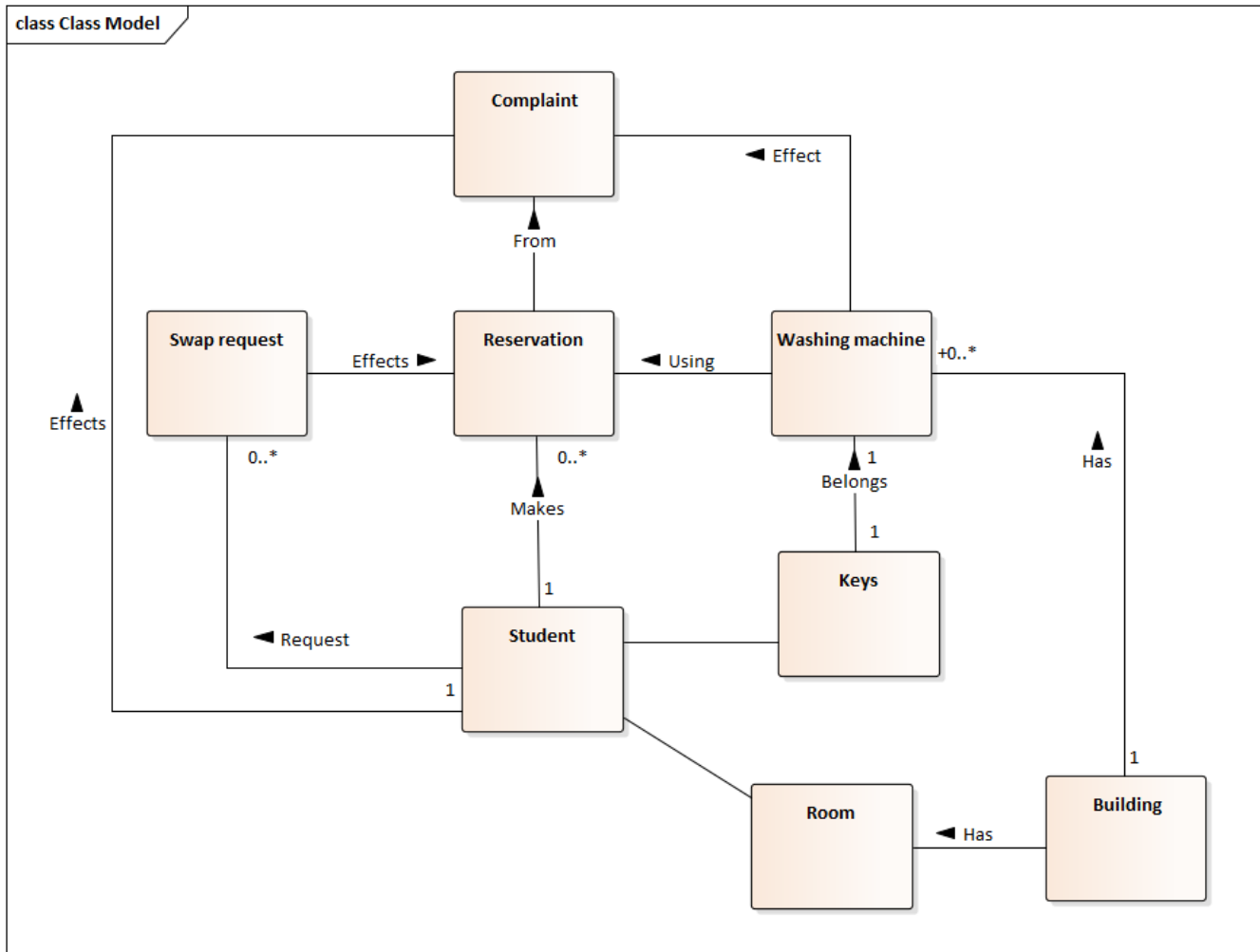


Figure 4 – Business Domain Model

3.1.1 Complaint

This entity represents the list of complaints sent by the washing machine users.

3.1.2 Reservation

The entity 'Reservation' holds reservations made by students for a particular machine for a specific time and day and for a chosen amount of time.

3.1.3 Student

Student represents a person that lives in the building and is need of use of the washing machines.

3.1.4 Swap request

The 'Swap request' entity is for students that have sent a request to another student for a taken time slot to be switched to them.

3.1.5 Washing Machine

A 'washing machine'.is the entity that represents one functional washing machine that can be used and reserved by a given user. Each machine has its own schedule that has time slots that can hold reservations.

3.1.6 Key

'Key' entity is a representation of the physical key that each machine needs for a user to access. The entity in the system is there so that an admin user can keep track of the last known location of any given key.

3.1.7 Room

The 'room' were the student user belongs to.

3.1.8 Building

The entity 'building' represents the building were a given number of users and machines are located. This entity gives the ability for one system to hold multiple subsystems of building with machines and users that are all functioning in an independent fashion.

4. Requirements

This section defines the requirements for the Dorm Services Key Rental System, divided into non-functional and functional requirements.

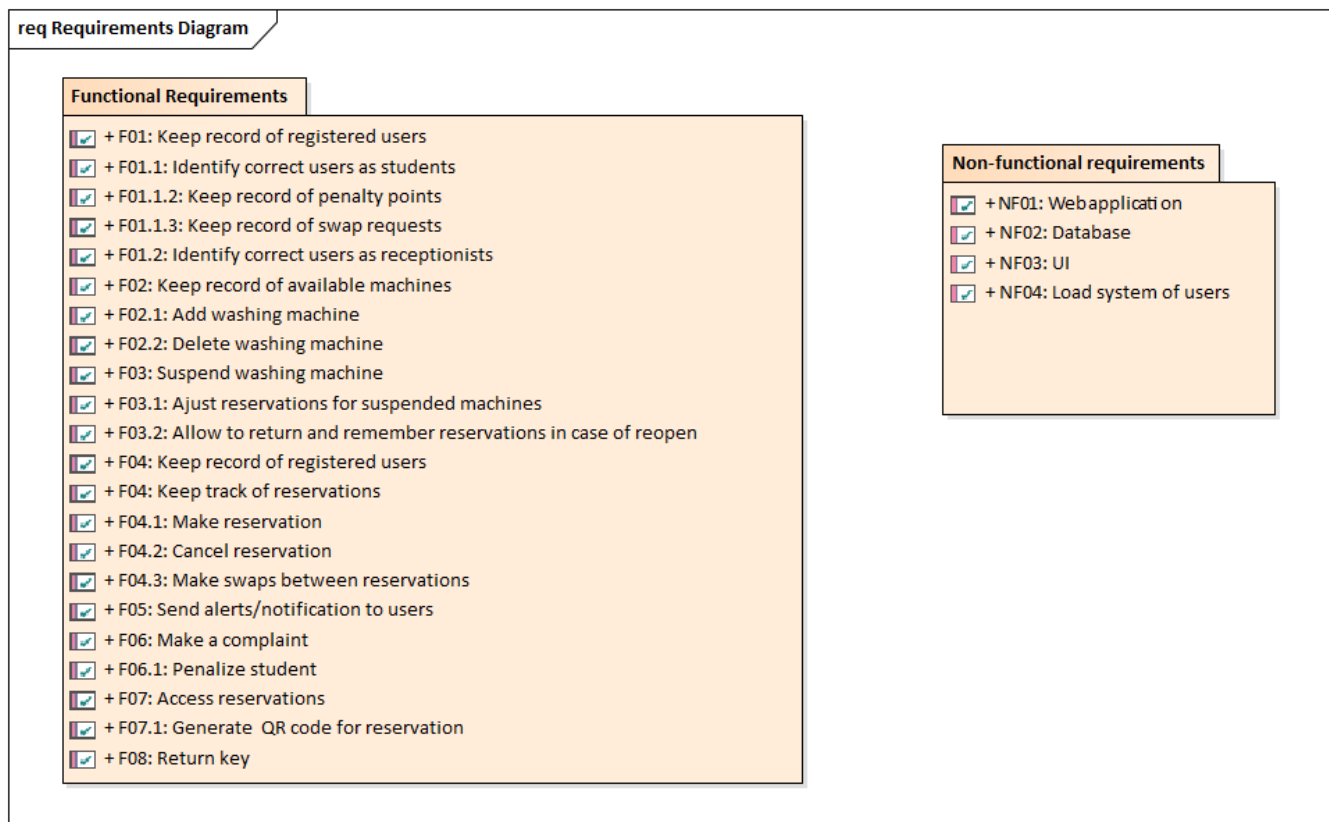


Figure 5 - Requirements Diagram

4.1 Non-functional requirements

In this section, the non-functional requirements of the Dorm Services Key Rental System are listed and described.

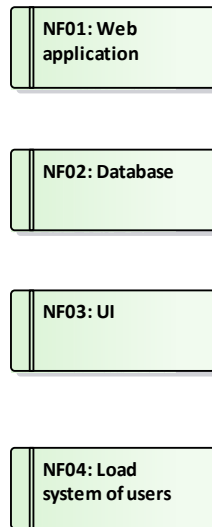


Figure 6 - Non-functional requirements

4.1.1 NF01: Web application

Priority: High

The system is a web application that will be accessed by the students and the receptionists. All features of the system will be accessible through this web application.

4.1.2 NF02: Database

Priority: High

The system must use a database to keep track of the students, the reservations, the washing machines and additional information. All this data will be available for the receptionist, but partially available for the students.

4.1.3 NF03: UI

Priority: High

The system needs a UI for the web application. The UI must be intuitive so the users (students or receptionists) can access all the features of the system easily.

4.1.4 NF04: Load system of users

Priority: High

The ability to load information that only the student should have access to. This is done in order to separate the data between the user and the receptionist.

4.2 Functional Requirements

In this section, the functional requirements for the Dorm Services Key Rental System are specified.

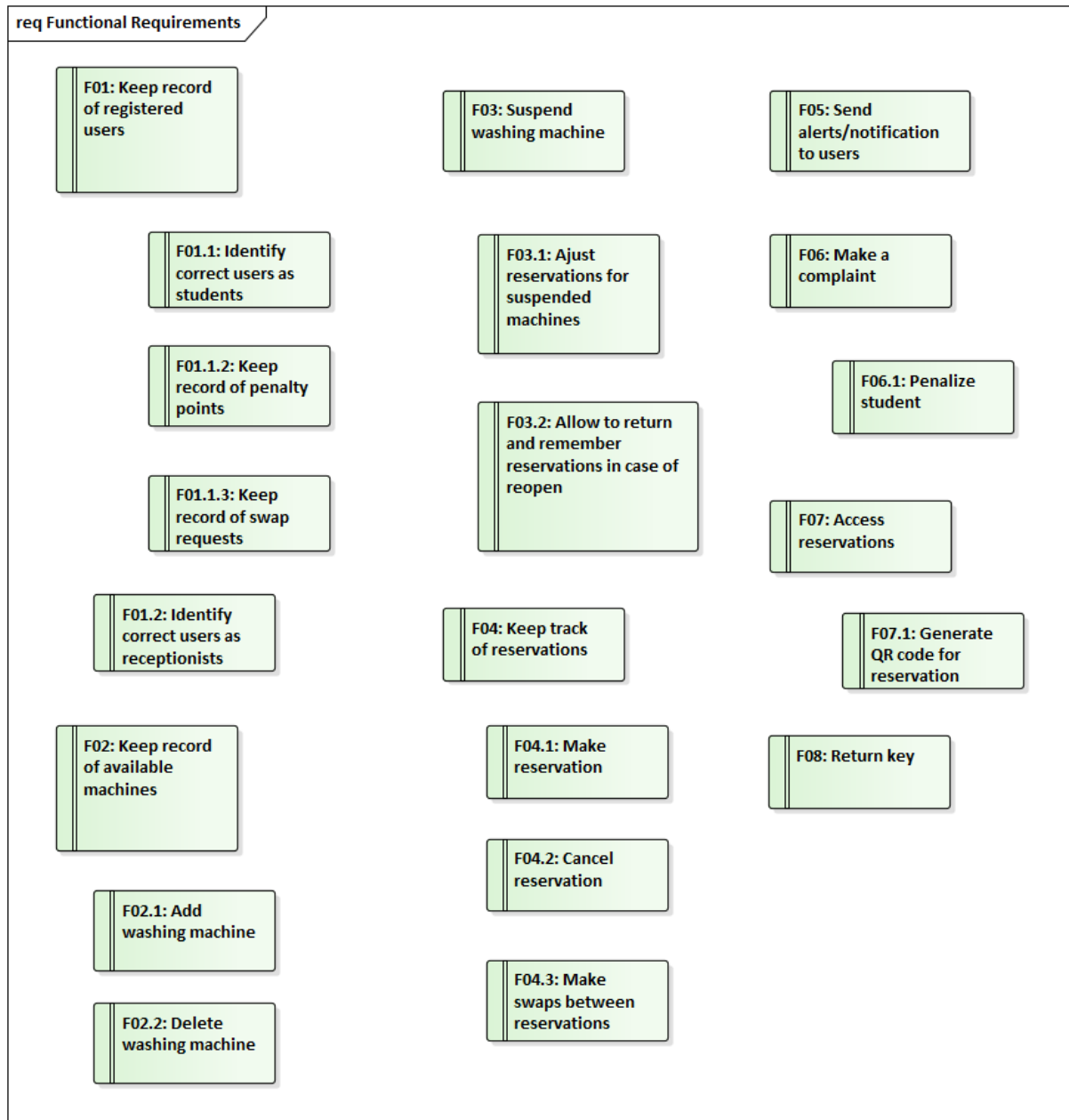


Figure 7 - Functional Requirements



4.2.1 F01.1.2: Keep record of penalty points

Priority: Medium

We need to keep record of penalty points each student has and at what point they can reset themselves. Students lose penalty points if they break any rule.

4.2.2 F01.1.3: Keep record of swap requests

Priority: Low

Swap requests should be tracked until the time they are no longer relevant (the requested swap time has passed) or until the student that holds the requested time has either accepted or rejected the request.

4.2.3 F01.1: Identify correct users as students

Priority: High

The system needs to be able identify specific users as student users by there credentials. These users should only have access to their own reservations, information, and alerts.

4.2.4 F01.2: Identify correct users as receptionists

Priority: High

The system must be able to Identify the correct users when they log in as receptionists and make sure to give them the correct account.

4.2.5 F01: Keep record of registered users

Priority: High

The system must keep a record of registered users to check if the person that is trying to log in is a student or not. Only students are allowed to access the booking system.

4.2.6 F02: Keep record of available machines

Priority: High

The system must keep a record of available machines to know if they are able to be used or not. With this information, the schedule will mark each washing machine as available or unavailable.

4.2.7 F02.1: Add washing machine

Priority: High

The receptionist should be able to add working washing machines to the system. To do that, the receptionist would need the number of the washing machine and the room where it is located.

4.2.8 F02.2: Delete washing machine

Priority: High

The receptionist should be able to delete washing machines that need to be replaced from the system. To do that, the receptionist would need the number of the washing machine and the room where it is located.

4.2.9 F03: Suspend washing machine

Priority: High

The ability to suspend a washing machine is used in the case of a problem or maintenance work in said machine. It means that the machines reservations for the selected amount of time are deleted, and the machine is considered temporarily closed for that time. To suspend a washing machine, the receptionist would need the number of the washing machine and the room where it is located.

4.2.10 F03.1: Adjust reservations for suspended machines

Priority: Low

If a machine is suspended the receptionist who suspended it must choose an approximated time for closer of the machine. All reservations during that time must be deferred or removed to a later time. The suspended time can be lengthened or shortened depending on the need.

4.2.11 F03.2: Allow to return and remember reservations in case of reopen

Priority: Low

If the amount of time chosen for the machine to be suspended is longer then required, a receptionist can choose to shorted it. If the time is shortened the reservations for that time should be returned.

4.2.12 F03: Keep track of reservations

Priority: High

The system must keep a record of reservations to create the schedule. With this schedule each user will be able to look for free time slots to make reservations.

4.2.13 F04.1: Make reservation

Priority: High

The student can make a reservation for a washing machine for a given amount of time. He has to find a free slot in the schedule and have enough penalty points to successfully acquire the requested reservation.

4.2.14 F04.2: Cancel reservation

Priority: High

The user and the receptionist will be able to cancel a reserved booking. In order for a student to do that, they need to have reserved a particular time slot. The receptionist can cancel reserved time slots from all students. If the receptionist cancels a time slot, they should provide a comment with the reason for cancelation.

4.2.15 F04.3: Make swaps between reservations

Priority: Low

The student can suggest making swaps between reservations to other students.

When a student requests a swap, the other student will be notified through email. The other student will be able to accept or decline the request through email. If the student accepts the request, the students will change their reservation times.

4.2.16 F05: Keep record of registered users

Priority: High

We need to keep record of registered users to check if the person that is trying to login is a student or not. Only students are allowed to enter the system.

4.2.17 F06: Send alerts/notification to users

Priority: Low

When a receptionist makes changes that effect a given users the said user should be notified. Also the user should be able to get notified by the system, when a washing machine is available for use. In order for this to happen, the student must set an alarm for a particular time slot. Alarms are possible to set every 30 min and the maximum amount of alarms per reservation is 5. Also, the swap requests and their response will be notified to the user.

4.2.18 F06: Make a complaint

Priority: Medium

A user should be able to make a complaint during there reservation time about a machine or a previous user, which is then sent to a receptionist.

4.2.19 F06.1: Penalize student

Priority: Medium

A complaint can result in a penalization of student, in which case the students penalty points should be effected and the complaint should be stored.

4.2.20 F07: Access reservations

Priority: High

When a user's reservation time starts the system should allow the user to access there reservation. The receptionists has access to all the reservations, but the students only to the one they have made.

4.2.21 F07.1: Generate QR code for reservation

Priority: Low

To access any machine in the given system a user would need to authenticate their reservation with the receptionist. This is done by using a QR code which is generated for the said reservation.

4.2.22 F08: Return key

Priority: High

When a student returns the key, the receptionist has to update it in the system. In order to do that, the receptionist will need the student ID.

5. Use Case Diagrams

In this section, the use cases of the Dorm Services Key Rental System are specified. The use cases define the functionalities available to users interacting with the system.

The following cases are considered crucial:

- Select and reserve/delete a reservation
- Show a list of all available time slots
- Access QR code/check in
- Activate usage of service
- Confirm service
- Close service

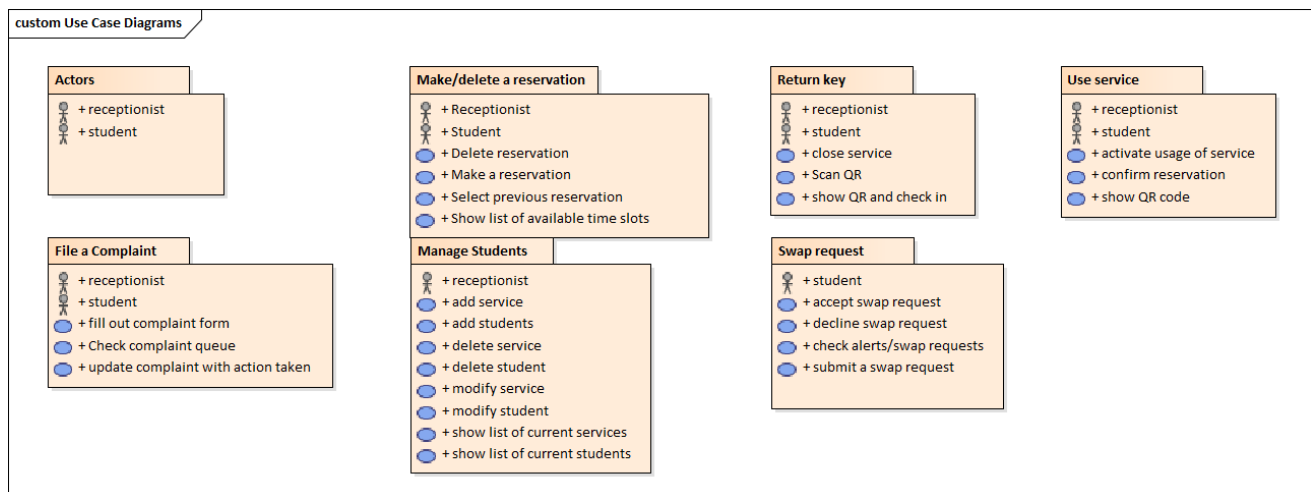


Figure 8 – Use Case Model

5.1 Actors

This package describes the actors of the system.

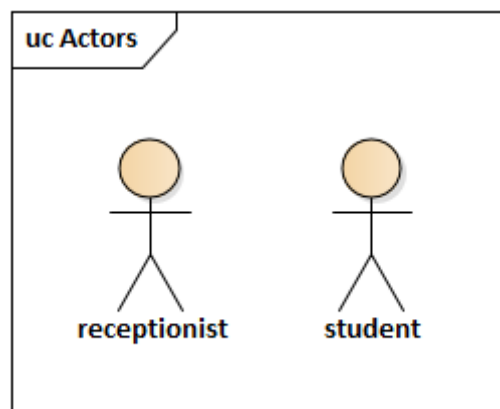


Figure 9 - Actors

5.1.1 Receptionist

A person working in the dorm with access to the key rental desktop web application.

Using the application, the receptionist manages key rental, student complaints, and penalizing students for misuse of facilities. They can find and change information about the students and their reservations.

5.1.2 Student

A person who wants to rent a key to use the dorm facilities. They can make/delete/swap reservations, file a complaint, and use their QR code to obtain a key.

5.2 File a Complaint

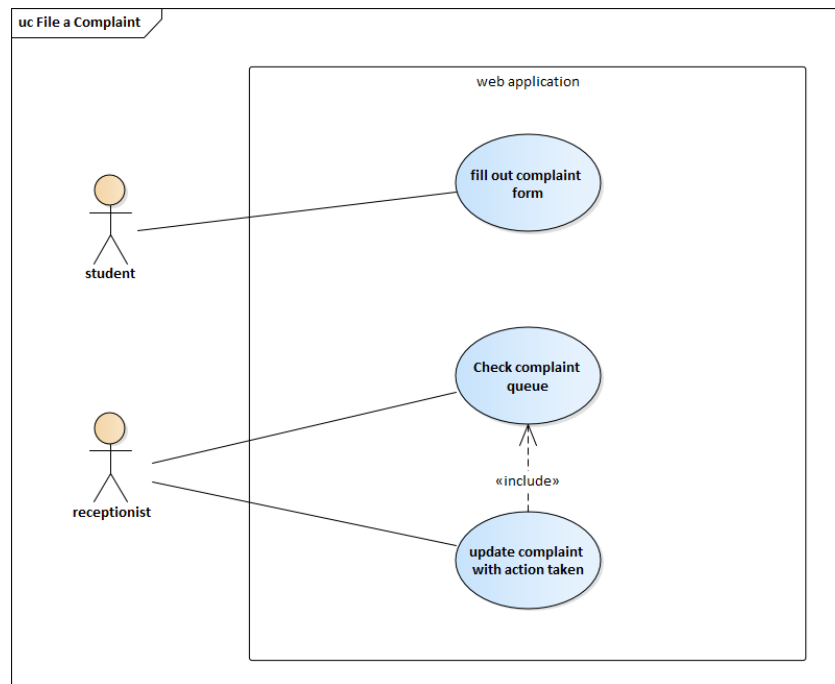


Figure 10 - File a Complaint

5.2.1 Fill out complaint form

The student may fill out a complaint form by specifying their name, type of issue, and complaint comments.

Basic Path: Student complaint

This scenario describes the basic process of a student complaint.

1. The use case starts when a student has an issue with the facilities/washing machine. The student creates a new complaint in the web application.
2. The system shows the complaint form with the following fields: name of user, type of issue, and comments. All of the fields are mandatory. The student also may upload a photo of the issue.

3. The student fills in the required fields and clicks the submit button.
4. The system validates the provided information and redirects the student back to their profile home.

5.2.2 Check complaint queue

The receptionist may check the complaint queue, where they can view the student complaints form all students in a dorm. The queue appears blank if there are no open issues. The receptionist may also access the history of closed complaint forms.

5.2.3 Update complaint with action taken

The receptionist may update the status of a complaint and make a note with the action taken. This will notify the person who made the complaint and any other users impacted because of the complaint.

Basic path: Update complaint with action taken

1. After checking the complaint queue, the receptionist may select a complaint and take action based on the issue.
2. The receptionist will take action by initiating repairs for the facilities and/or by penalizing students who misused the facilities.
3. The receptionist may change the status of the issue to closed and make a note describing what action (if any) has been taken. Both fields are required.

Exception: No action taken

1. If no action is taken, the receptionist may make note that no action has been taken and close the issue.

5.3 Make/delete a reservation

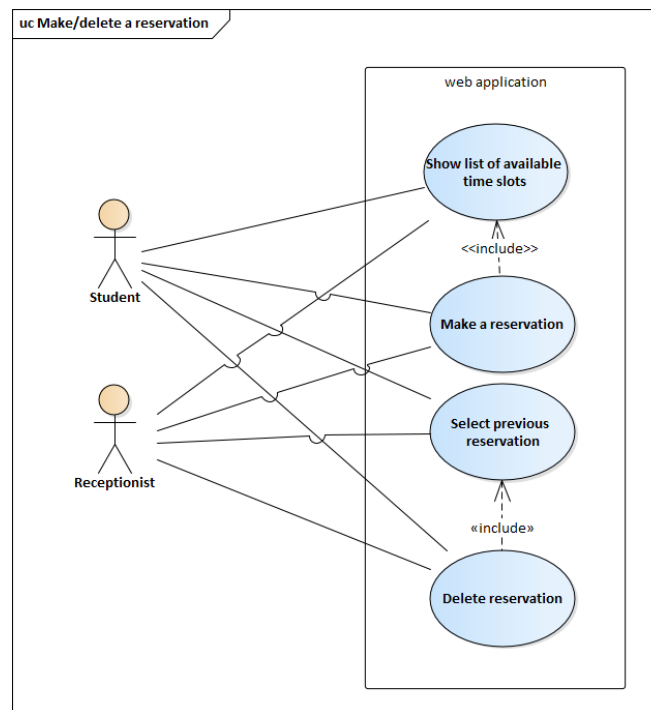


Figure 11 - Make/delete a reservation

5.3.1 Show a list of available time slots

Both the receptionist and the student may view the list of available time slots.

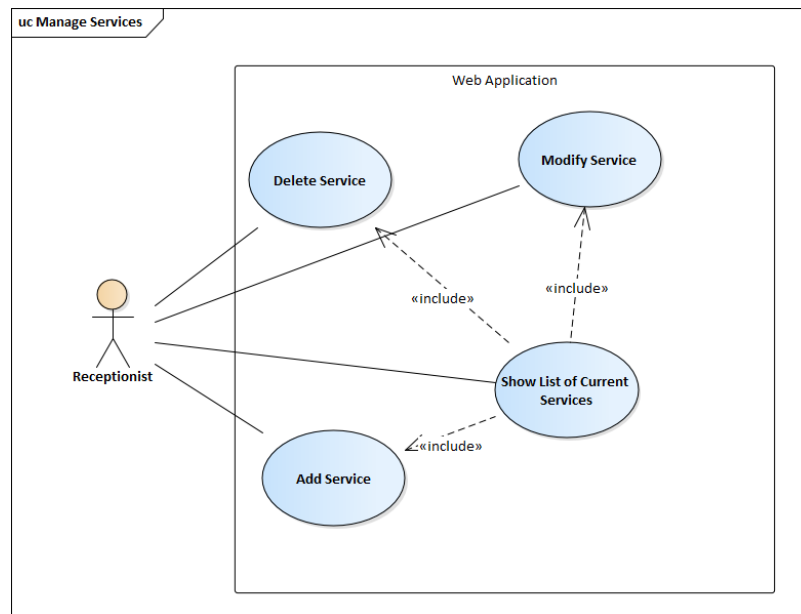
5.3.2 Select and reserve/delete reservation

Both the receptionist and the student may select and reserve or delete a reservation. Students may only create and delete reservations for themselves. The receptionist may create and delete reservations for all students in the dorm system.

Basic path: Create a reservation

1. The receptionist or student will look at the timetable and select a time slot. (The receptionist view will allow selection of current bookings or empty time slots. The student view will only allow for selection of empty time slots or slots they have already reserved.)
2. If an empty slot has been selected, the student can make a reservation, or the receptionist can make a reservation for a student.
3. If a slot that has already been reserved is selected, the receptionist or student may delete it.

5.4 Manage Services/Students



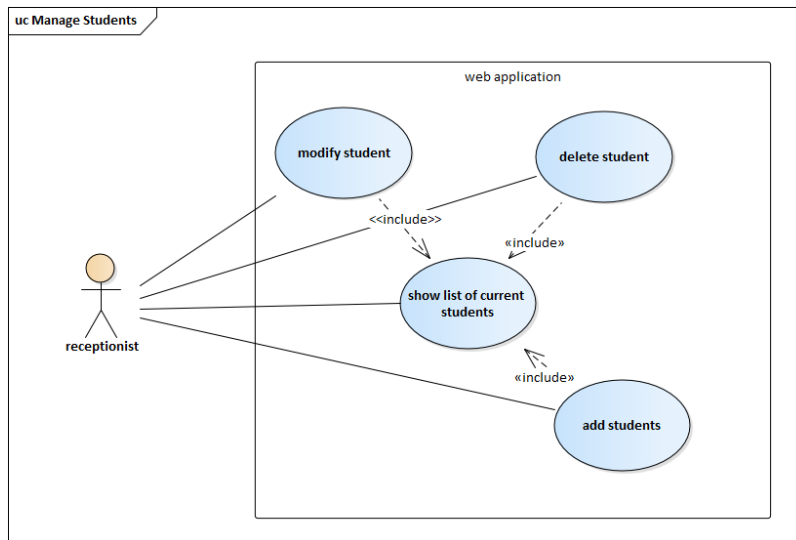


Figure 12 - Manage Services/Students

5.4.1 Add/delete service

Laundry reservation is the default service. However, the receptionist may add or delete a dorm facility/service that students may be able to create a reservation for (e.g. a student gym, study room, etc.). The receptionist will be required to enter the name of the service and how many items of the service are available at a given time (e.g number of washing machines, gym/study room capacity, etc.).

Basic path: Managing services

1. From the managing services view, the receptionist may view a list of current services.
2. Selecting a service will allow the receptionist to modify values of the service or delete the service.
3. The receptionist may also select an option to add a new service.

5.4.2 Show a list of current services

The receptionist will be able to view a list of the current services in the system.

5.4.3 Modify service

The receptionist will be able to delete a service or modify the number of items of the service are available at a given time. This will be useful, for example, if a machine is out of order or if a room has limit capacity due to COVID.

5.4.4 Modify student

The receptionist will be able to modify any fields in a student profile.

5.4.5 Add/delete students

The receptionist will be able to add and delete student profiles.

Basic path: Managing students

1. The options for managing students are the same as managing services. A student profile may be deleted, created/added, or modified.

5.4.6 Show a list of current students

The receptionist will be able to view a list of current students. It is from this view that the receptionist may select an option to delete or modify a student profile.

5.5 Return key

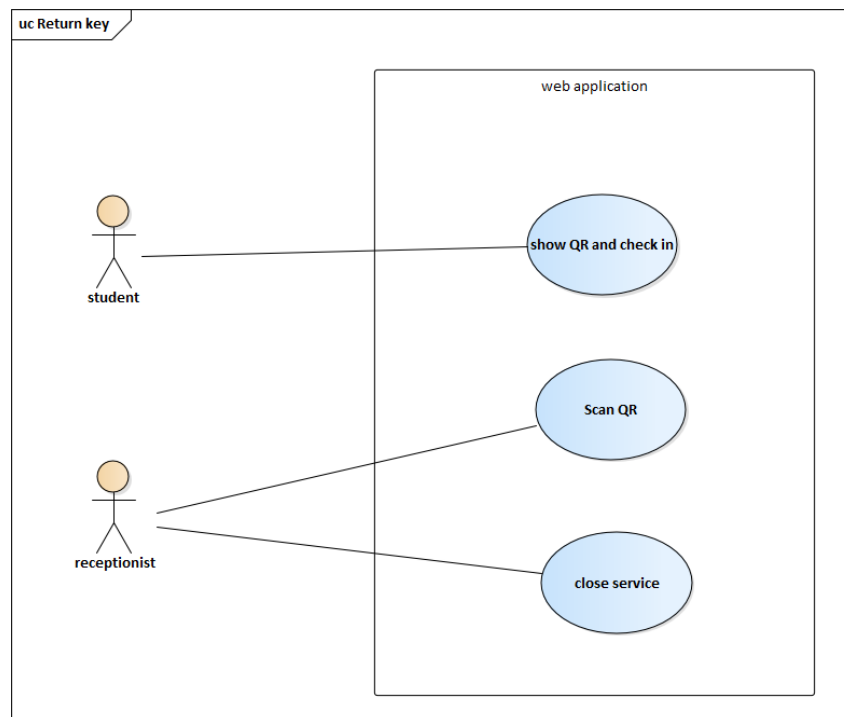


Figure 13 - Return key

5.5.1 Access QR and check in

The student will be able to access their unique QR code identifier from their student profile and use it to check in with the receptionist when they present their key at the end of a reservation.

Basic path: Returning a key

1. A student will access their QR code from their profile and show it to the receptionist.
2. The receptionist will scan the QR code. When the QR code has been scanned, the reservation will be closed and it will be marked that the key has been returned.
3. The student will return the key.

Exception: Key not returned

1. If the student fails to return the key, the reservation will be closed when the time of the reservation ends and it will be marked that the student never came to return the key in the system.

5.5.2 Close service

The receptionist will be able to scan the QR code of the given student. If the student has an open reservation, the reservation will be closed, and it will be noted that the key has been returned.

5.6 Swap request

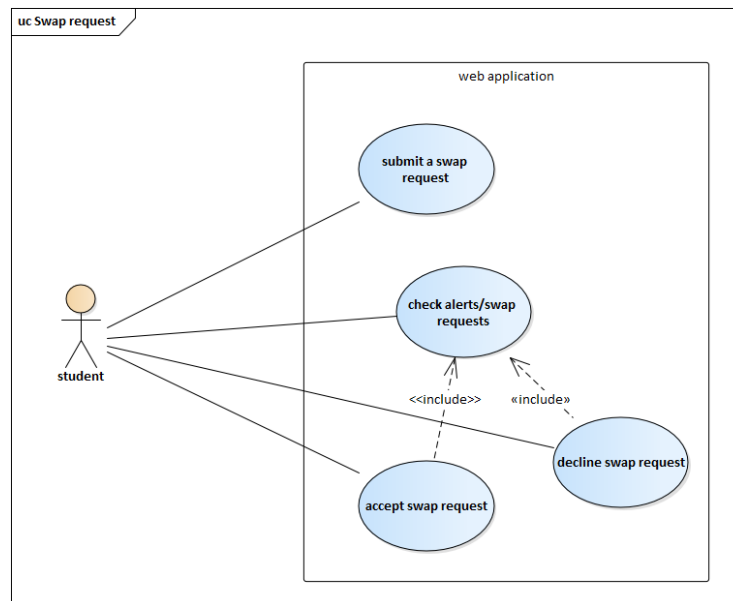


Figure 14 - Swap request

5.6.1 Submit a swap request

If another student has a reservation during a time desired by Student1, they may select that time and submit a swap request.

Basic path: Swapping a time slot with another student

1. If a student would like to swap a time that another student has reserved, they may select that time and their own reservation they want to swap with and submit a request to swap.
2. The other student who the request was sent to may check their alerts for swap requests and they may accept or decline the request.
3. If the request has been accepted the two reservations will be swapped between the students.

5.6.2 Check alerts/swap requests

Students may check a list of alerts and swap requests in the system.

5.6.3 Accept swap request

If a student wishes to accept a swap request, they may select accept. Otherwise, they select decline. Student1, who submitted the request, will be notified if the request has been accepted or declined.

5.7 Use service

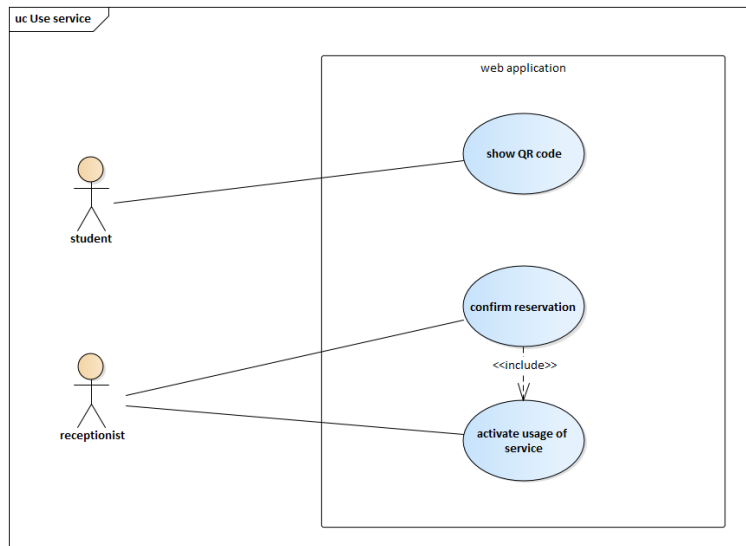


Figure 15 - Use service

5.7.1 Show QR code

A student will be able to access their unique QR code identifier from their profile page.

Basic path: Swapping a time slot with another student

1. A student will access their QR code in their profile and show it to the receptionist.
2. The receptionist will scan the QR code and confirm that the student has a reservation. When the QR code is scanned, the student's reservation will be activated and it will be marked that the student received a key.
3. If the student has a reservation, the receptionist will give the student the key.

5.7.2 Confirm reservation

The receptionist can scan the student's QR code and confirm that the student has an existing reservation. The receptionist may then give the student the key required to use the service.

5.7.3 Activate usage of service

Once a reservation has been confirmed, the system will mark the reservation as active and take note that the student has collected the key from reception.

6. Domain model

This section describes the domain of the dorm laundry system, defining the properties and relations of the important objects of the domain.

The main entity

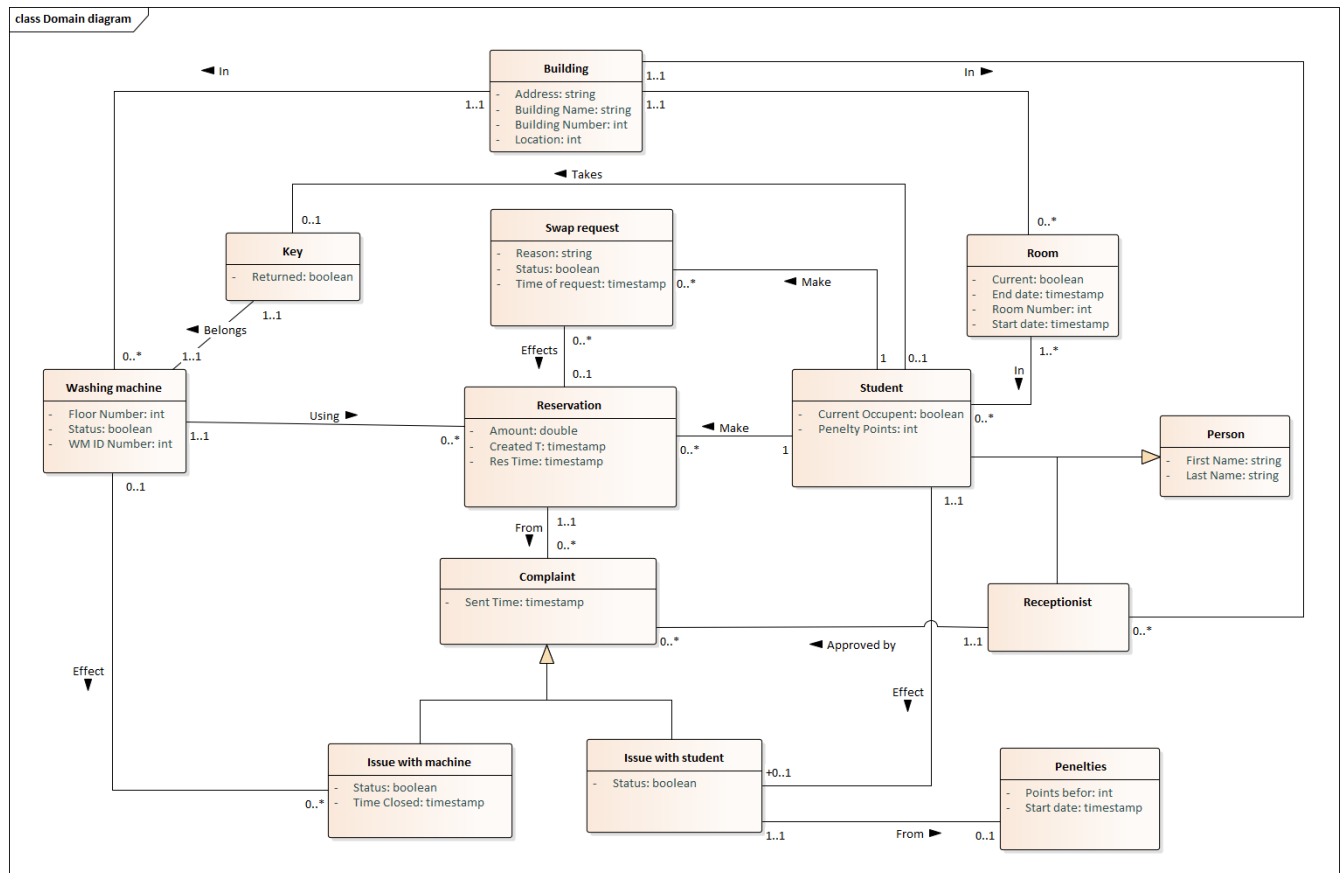


Figure 16 - Domain diagram

6.1.1 Complaint

This entity represents the list of complaints sent by the washing machine users.

| Attribute title | Description |
|-----------------|---|
| Sent Time | The timestamp of when the issue was realized and sent in. |

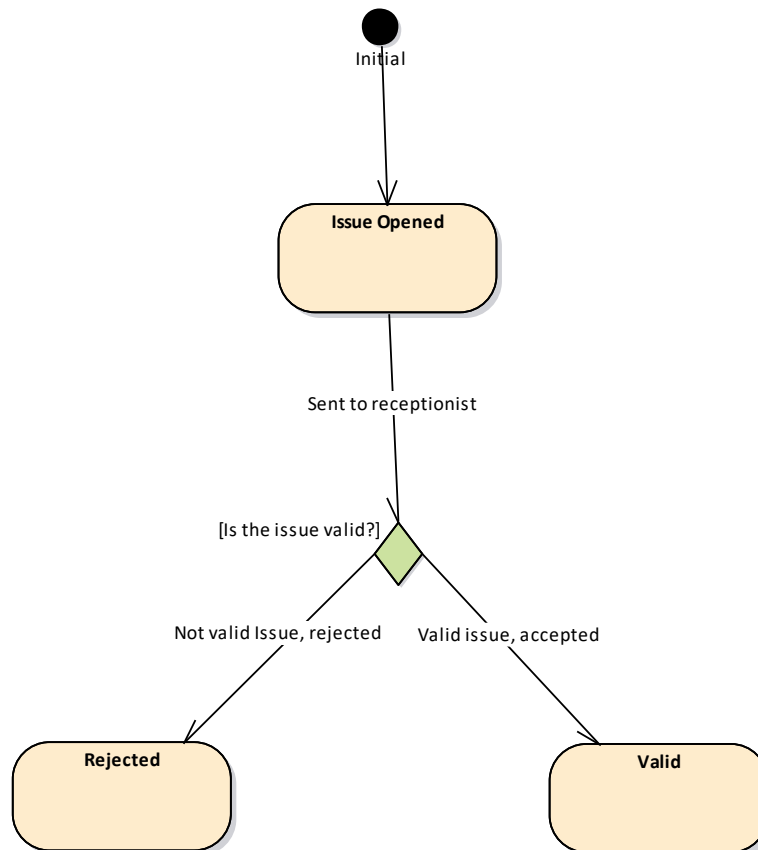


Figure 17 - Issue State Model

This diagram shows the states of the complaints.

When a complaint is made it is sent directly to the inbox of all relevant receptionists.

Once the receptionist responds to the complaint the complaint is either moved to rejected, meaning it is thrown out and no longer considered relevant (not enough evidence or not at a level that it should be addressed).

Or the complaint is moved to valid where it is addressed and stored for reference.

6.1.2 Issue with machine

This entity is to represent validated issues of washing machines that effect the schedule of a given machine.

| Attribute title | Description |
|-----------------|---|
| Status | The status of the complain of whether it was considered valid or not (if seen). |
| Time Closed | The time the issue was seen and checked. |

6.1.3 Issue with student

This entity holds the information of complaints that have been approved by a receptionist as valid with all the information about the issue.

| Attribute title | Description |
|-----------------|--|
| Status | Whether the complaint was seen and answered. |

6.1.4 Receptionist

Receptionist is an entity that represents a person with a management account.

6.1.5 Reservation

The entity 'Reservation' holds reservations made by students for a particular machine for a specific time and day and for a chosen amount of time.

| Attribute title | Description |
|-----------------|---|
| Amount | The duration time of the reservation. |
| Created T | The time when the reservation was created. |
| Res Time | The time when the start of the reservation was created for. |

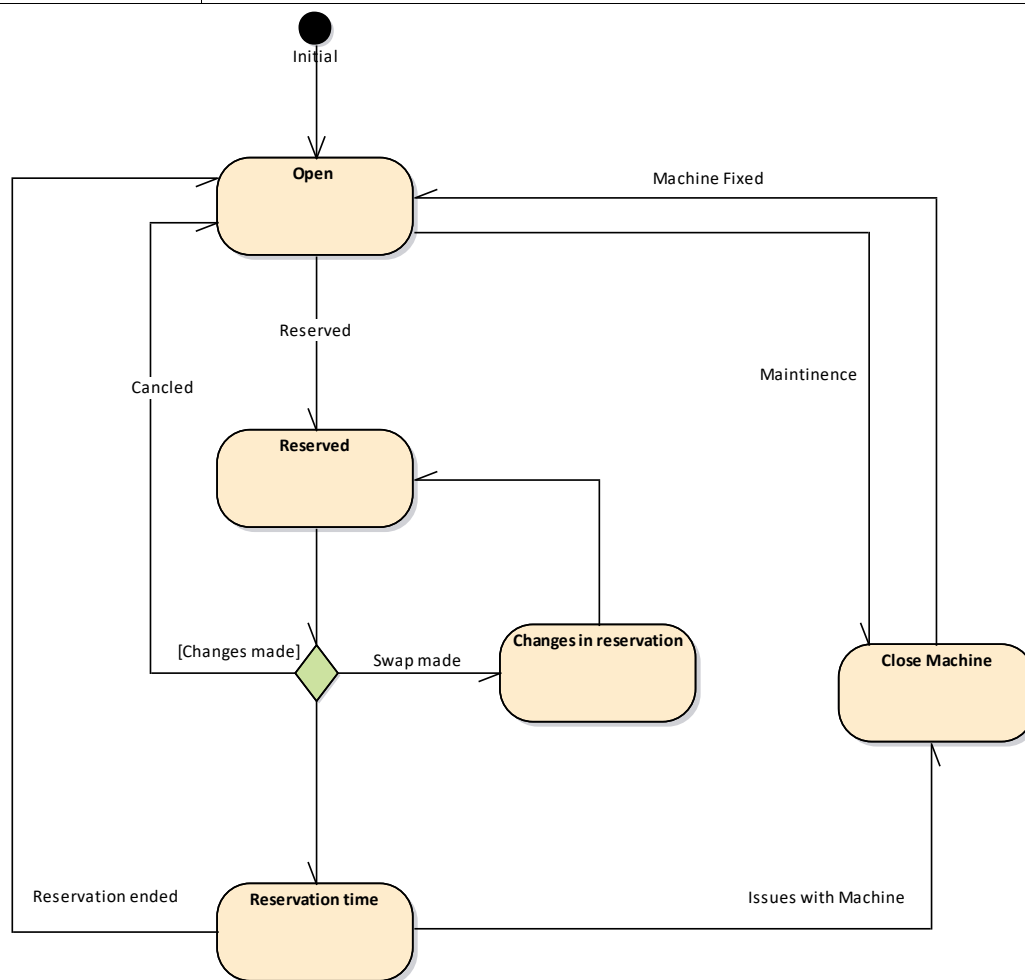


Figure 18 - Reservation State Model

This diagram represents the states of the timeslots for the machine. When a timeslot is free it is considered open and available for a student to take or for the timeslot to be closed to the machine being in maintenance or closed for good. When a student makes a reservation for that time the state moves to reserved. From there it can be changed to another student if a swap is agreed upon.

Canceled if the student or receptionist chooses to, which would move the timeslot to the open state, or the student can continue to the reservation time when it begins.

Once the time is finished the machine can be reopened for the next timeslot or the machine can be closed for maintenance or for good due to an issue with the machine.

6.1.6 Student

Student represents a person that lives in the building and is need of use of the washing machines.

| Attribute title | Description |
|------------------|---|
| Penalty Points | The amount of point a specific student holds at that exact moment. |
| Current Occupant | This Boolean represents if this user is currently a occupant in the location. |

6.1.7 Swap request

The 'Swap request' entity is for students that have sent a request to another student for a taken time slot to be switched to them.

| Attribute title | Description |
|-----------------|--|
| Status | Whether the swap was accepted or not. |
| Reason | The reason for the request of the swap. |
| Time of Request | The timestamp of the time the request was sent |

6.1.8 Person

An entity of a 'person' represents a user of the application

| Attribute title | Description |
|-----------------|---|
| First Name | The First name of the user that is stored when they register. |
| Last Name | The last name or surname of the user that is stored when they first registered. |

6.1.9 Washing Machine

A 'washing machine' is the entity that represents one functional washing machine that can be used and reserved by a given user. Each machine has its own schedule that has time slots that can hold reservations.

| Attribute title | Description |
|-----------------|--|
| Floor Number | The floor in the given building that the machine is located in. |
| Status | The status of whether the machine is functioning at that moment and is accepting reservations. |
| WM ID | The unique ID number of the mashing machine. |

6.1.10 Key

'Key' entity is a representation of the physical key that each machine needs for a user to access. The entity in the system is there so that an admin user can keep track of the last known location of any given key.

| Attribute title | Description |
|-----------------|---|
| Returned | The Boolean status of whether the key is located at the main office of the building it belong in. |

6.1.11 Room

The 'room' where the student user belongs to.

| Attribute title | Description |
|-----------------|---|
| Room Number | The room number in a given building. |
| Current | If the room to student connection is the on that is current. |
| End time | The time, if ever, the student no longer resides in that room |
| Start time | The time the student began residence in the room. |

6.1.12 Building

The entity 'building' represents the building were a given number of users and machines are located. This entity gives the ability for one system to hold multiple subsystems of building with machines and users that are all functioning in an independent fashion.

| Attribute title | Description |
|-----------------|---|
| Address | The address of the building. |
| Building Name | The name of the building, if any. |
| Location | The number representation of the location in the system in order to be able to have multiple simultaneous systems in one environment at once. |
| Building Number | The unique ID number of the building in the system. |