

# Use cases for Roskilde daycare

**Scope:** Roskilde Daycare Program

**Assumptions and pre-conditions for all use-cases:**

- *Open between 8am and 5pm, Monday to Friday.*
- *The daycare has one computer, accessed by one person at a time.*
- *One admin and maximum of 5 other employees, some of which are SuperUsers.*
- *All children have at least one parent.*
- *Daycare has a max of 15 children registered.*
- *Children move from waiting list to active list based solely on registration date (FIFO).*
- *Parents must be created first. All parents must have basic information to be able to register.*
- *All Employees must have basic info to be able to register.*

**Technology and data variations list for all use-cases:**

- *Basic PC*
- *Printer*

## **Manage work schedule (Casual use-cases)**

### ***Create a new Schedule.***

**Primary Actor:** *Admin*

**Preconditions:**

- *Terminal is logged in with admin credentials*
- *No schedule exists for the chosen week*

**Postcondition:**

- *A new schedule is created*

**Main Success Scenario:**

- *Admin creates new schedule for the desired week*
- *The admin assigns certain employees to work on specific days*
- *The admin saves and sends out the schedule*
- *The admin logs in to the system*

**Extension:** Invalid week

- *The week chosen already has a schedule. The admin is prompted to change the week number until it's valid*

---

## **Update Schedule**

**Primary Actor:** *Admin*

**Preconditions:**

- *Terminal is logged in with admin credentials*
- *A schedule exists for the chosen week*
- *Information on the schedule is deemed outdated*

**Postconditions:**

- *Information on schedule has been changed*

**Main Success Scenario:**

- *Admin chooses specific schedule*
- *Admin then updates info for the affected day/days*

**Extension: Clashing information**

- *The admin tries to assign hours to an employee who is either already working that day, or on an invalid day*

---

## **View Schedule**

**Primary Actor:** *Employee/User*

**Preconditions:**

- *Terminal is logged in with employee credentials*
- *At least 1 schedule exists*

**Postcondition:**

- *A schedule gets printed to the console*

**Main Success Scenario:**

- *Employee chooses the view schedule option*

- *The schedule is displayed*

**Extension: No schedules exist**

- *The system displays an error message informing the user that no schedule exists*
- 

**Delete schedule**

**Primary Actor:** *Admin*

**Preconditions:**

- *Terminal is logged in with Admin credentials*
- *At least 1 schedule exists*

**Postcondition:**

- *A schedule has been deleted*

**Main Success Scenario:**

- *Admin chooses a schedule to delete*
- *The system prompts the admin for a confirmation*
- *The admin confirms and the schedule is deleted*

**Extension: Cancel**

- *The admin is prompted for a confirmation but chooses to cancel. The system boots them to the schedule menu*
- 

**Managing children's information (Fully dressed use-case):**

**Primary actor:** *Admin, Employee/user*

**Stakeholders and interests:**

- *The admin - want a fast, simple, reliable System to work on*
- *The admin wants the system to be able to access/change/create(modify) information about children in the daycare*
- *The Parents wants the employees to be informed about information concerning their children*
- *The employees want to be able to search information about a child*

**Success guarantee (Postconditions):**

- *Information modified is correctly saved to the database and accessible*

**Special requirements:**

- *None*

**Main success scenario (Admin):**

- *The admin logs in using his/her credentials*
- *The system confirms login*
- *The system shows Admin menu*
- *The admin selects create child*
- *The admin creates a child into the system with first name, last name, CPR, the first day in daycare etc.*
- *The new child is saved into the database*
- *The system goes back to the menu*
- *The admin quits the program once done*

**Main success scenario (Employee/user):**

- *The employee/user logs in using his/her credentials*
- *The system confirms login*
- *The employee/user selects search child*
- *The employee/user types in CPR*
- *The system shows the information added in the database of the child*
- *The system goes back to the main menu*
- *The employee quits the program once done*

**Alternate flows:**

- *The information about the children and parents are wrongly saved in database*
- *Information has changed from what was previously saved in the database*
- *The System or computer used is down*

---

**Manage employees (Fully dressed use-case)**

**Primary Actor:** *Employee/user*

**Stakeholders and interests:**

- *All employees/users – want a fast, simple, reliable System to work on*

- *Children – need the System to provide a safe, reliable environment so they are able to have their needs met in terms of staffing hours etc.*
- *Parents – want their info and payments kept up to date, and their children to be adequately staffed*

**Preconditions/assumptions:**

- *There is one Admin, and the rest of the staff are Users/Superusers*

**Success guarantee/postconditions:**

- *Employee is created and able to access the System*

**Main Success Scenario:** *[next section]*

**Extensions/alternative flow:** *[next section]*

**Special requirements:**

- *None*

**Main success scenario/alternative flow:**

- *Admin logs on to System using credentials*
  - *System shows confirmation of login*
  - *If login information is incorrect, System displays prompt to re-enter*
  - *Admin selects 'create roster' from Staff Management Menu*
  - *Admin selects week to create roster for*
  - *Admin selects people to work from list of current Employees*
  - *System prompts whether information is final and correct and shows summary*
  - *Admin confirms*
  - *If Admin declines, the system reverts back to previous screen*
  - *System goes back to Staff Management Menu*
- 

## **Register Payment (Fully dressed use-case)**

**Primary Actor:** *Admin*

**Stakeholders and interests:**

- *The employees want to have their salaries on time*
- *The parents want to be assured that their kid's financial obligations are met*
- *The Admin wants the payments to be prompt without delays*

**Preconditions/assumptions:**

- Only the Admin can handle payments
- The payments are monthly

**Success guarantee/postconditions:**

- *A payment is registered in the system*

**Main Success Scenario:** *[next section]*

**Extensions/alternative flow:** *[next section]*

**Special requirements:**

- *None*

**Main success scenario/alternative flow:**

- *Admin logs on to System using her credentials*
- *System shows confirmation of login*
- *If login information is incorrect, System displays prompt to re-enter*
- *Admin selects Register Payment from Payments Menu*
- *Admin selects the child whose parents have paid*
- *If an inexistent child is selected, the System prompts for re-entry*
- *System asks Admin to enter the payment*
- *Admin enters the payment*
- *System asks for confirmation*
- *Admin confirms*
- *If Admin declines, the system reverts back to previous screen*
- *System goes back to Payments Menu*