**D:\Work\RoboOfficeNXT\Artwork\RoboOfficeNXT.pngRoboOfficeNXT ERP User Cases**

|  |  |  |
| --- | --- | --- |
| Title | Create student | |
| Description | A new student want to register, office process the registration process. | |
| Primary Actors | Logged in Office Receptionist | |
| Entry Conditions | Admission form filled  Optional: Assessment form | |
| Exit Conditions | A new student is created in the system | |
| Flow of Events | 1. Receptionist press “new student” 2. Receptionist fill out student information according to the admission form 3. Receptionist fills out student assessment information according to the assessment form. 4. Receptionist press “Save”   9) Receptionist selects an option, proceeds to the next User Case | 1. System displays the “Add New Student” Tab, that includes all text fields on admission form   6) System does field-check on all information entered.  7) System attempts to store information into database  8) If successful, system logs the creation action by logged in user.  displays student number and show list of options  -- Course Payment?  -- Membership/Club Payment?  -- Pay other fees?  -- Prospective student |
| Alternate Flow | 4a) Receptionist press “Cancel”  6a) Receptionist confirms, tab is closed | 5a) System prompt “Are you sure you want to discard information” |
| Special Requirements | None | |

**Student Management**

|  |  |
| --- | --- |
| Title | Modify Student personal information |
| Description | Receptionist edits a student’s personal info |
| Primary Actor | Logged in Office Receptionist |
| Entry Condition | new student information |
| Exit Condition | Student information updated |
| Flow of event | 1. Receptionist opens a specific student 2. System load student records, everything that’s related to a student 3. Receptionist choose “Edit” in personal information panel 4. System display editable fields for the receptionist to edit 5. Receptionist updates new student information and press “Save” 6. System does a field check on every field. Prompt the user “Invalid fields” if necessary. 7. System updates student personal information in database 8. If successful, System loads the student information part again, display new student information. |
| Alternate Flow | 1. During Edit, Receptionist press “Cancel” 2. System discard new information, exit editing mode, displays the old student information. |
| Special Requirements | None |

|  |  |
| --- | --- |
| Title | Change student’s timeslot |
| Description | Student requests a time slot change to one of his/her course progress |
| Primary Actor | Logged in Office Receptionist |
| Entry Condition | Student has an active course progress that is not expired |
| Exit Condition | Time slot changed |
| Flow of event | 1. Receptionist opens the student’s record tab 2. System displays all active student course processes 3. Receptionist choose the desired course process 4. Receptionist checks the timeslot changed number to see how many times the student has changed timeslot 5. Receptionist click “Change Time” Button 6. System displays a Dialog that has all available time slots for the particular course (available as in classroom capacity not reached for that timeslot) 7. Receptionist choose a timeslot 8. System updates the timeslot of the course progress for database 9. System reloads the course progress from database |
| Alternate Flow | a) During choose timeslot, Receptionist can select “Cancel”  b) System exit editing timeslot mode. |
| Special Requirements | None |

|  |  |
| --- | --- |
| Title | Mark student absent/deletes student absent |
| Description | Receptionist get a call from student parent, need mark student absent on a particular day or date range |
| Primary Actor | Logged in Office Receptionist |
| Entry Condition | a particular student, a date range, and the student must have an active course progress |
| Exit Condition | Absent is marked for the given date range |
| Flow of event | 1. Receptionist open a student record, and select “Report Absent” 2. System displays the absent history of the student 3. Receptionist enter a single date, or date range 4. Receptionist press “Absent” 5. System creates a new absent list entry 6. System logs absent made by user 7. System reloads absent list, display it. |
| Alternate Flow | a) Store manager deletes a student absent  b) System removes that entry from the list |
| Special Requirements | None |

|  |  |
| --- | --- |
| Title | Modify Class Record |
| Description | Change single lesson record information for student |
| Primary Actor | Logged in Store Manager |
| Entry Condition | A class record exists for a student’s course process |
| Exit Condition | Class record updated |
| Flow of event | 1. Manager choose to view detail progress report 2. System displays “Student Timeline” 3. Manager clicks a particular Class record 4. System displays class record information in record panel 5. Manager choose to edit the class record 6. System enters editing mode   --attend date  --project  --delete this record? |
| Alternate Flow |  |
| Special Requirements | Store manager privilege |

|  |  |
| --- | --- |
| Title | View Weekly Schedule |
| Description | Receptionist view weekly schedule by calendar |
| Primary Actor | Logged in Receptionist |
| Entry Condition | None |
| Exit Condition | None |
| Flow of event | 1. Receptionist selects “View Weekly Schedule” 2. System find current week 3. System displays students satisfy the following condition   --Active student with active progress  --students who don’t have an absent date on the current date  --date that doesn’t have a holiday  Listing is by Classroom, then by day of week  4) Every time a receptionist changes weeks, the information is refreshed. |
| Alternate Flow | None |
| Special Requirements | None |

|  |  |
| --- | --- |
| Title | View Attendance |
| Description | Receptionist print an attendance for a particular day |
| Primary Actor | Logged in Receptionist |
| Entry Condition | A day is chosen |
| Exit Condition | Attendance |
| Flow of event | 1. Receptionist click View Attendance 2. System displays Calendar and associated timeslots. If it’s holiday, display nothing 3. Receptionist Click a timeslot 4. System displays all students in weekly schedule that associates with the timeslots. Student is ordered by classroom name. |
| Alternate Flow |  |
| Special Requirements | None |

|  |  |
| --- | --- |
| Title | Print Attendance (extends View Attendance) |
| Description | Receptionist print an attendance for a particular day |
| Primary Actor | Logged in Receptionist |
| Entry Condition | A day is chosen |
| Exit Condition | Attendance |
| Flow of event | 1. Receptionist click Print Attendance 2. System processes all students and export excel sheet 3. System opens excel attendance file   8) Receptionist click print in Excel |
| Alternate Flow |  |
| Special Requirements | None |

|  |  |
| --- | --- |
| Title | Enter Attendance |
| Description | Receptionist enters an attendance for a particular timeslot |
| Primary Actor | Logged in Receptionist |
| Entry Condition | An attendance is available |
| Exit Condition | Attendance for students showed up is entered in the database |
| Flow of event | 1. Receptionist clicks Calendar date and timeslot 2. System displays the list of students for weekly schedule of that particular time slot 3. Receptionist clicks the students that are actually here 4. Receptionist clicks “Save” 5. System updates class record for each student that is here. Mark class record as incomplete 6. System prompt that record complete, prompt duplicates if found 7. System closes the tab |
| Alternate Flow |  |
| Special Requirements | None |

|  |  |
| --- | --- |
| Title | Delete a student |
| Description | Store manager deletes a student, permanently |
| Primary Actor | Logged in Store Manager |
| Entry Condition | Student exists within the system |
| Exit Condition | Student is deleted and is no visible anymore within the system |
| Flow of event | 1. Store manager navigates to student’s profile 2. Store manager clicks “Delete” 3. System prompt user “Delete is permanent! Do you want to delete?” 4. System mark student as deleted flag true 5. System prompt user “Student Deleted” 6. System close the student profile 7. System refreshes student lists |
| Alternate Flow |  |
| Special Requirements | None |

**RoboTeacher**

|  |  |
| --- | --- |
| Title | Enter Student progress report/ Complete student class record |
| Description | Teacher completes student class record after office records attendance |
| Primary Actor | Logged in Teacher |
| Entry Condition | An incomplete class record is available |
| Exit Condition | Class record is complete |
| Flow of event | Teacher opens a student’s class record  System display incomplete information fields  Teacher fills project information  Teacher fills in progress report information/ test report information  Teacher clicks “submit”  System update class record, mark as complete |
| Special Requirements | None |

|  |  |
| --- | --- |
| Title | View past student course progress |
| Description | Teacher needs to see student class history |
| Primary Actor | Logged in Teacher |
| Entry Condition | the student is available in the system |
| Exit Condition | Class records history is displayed |
| Flow of event | Teacher opens a student  System displays student’s progress for all courses  Teacher can click into each class record to see pass comments, scores, projects |
| Special Requirements | None |

**Store Manager**

|  |  |
| --- | --- |
| Title | Add a course |
| Description | Store manager add a course into the system |
| Primary Actor | Logged in Store manager |
| Entry Condition | None |
| Exit Condition | Course is complete |
| Flow of event | 1. Store manager add a course 2. System display the setting interface 3. Store manager input course code, name, and description 4. Store manager chooses a classroom 5. Store manager sets a rate and class duration 6. Store manager chooses the timeslots for the course 7. Store manager chooses the skill sets for the course |
| Special Requirements | None |

|  |  |
| --- | --- |
| Title | Add a project |
| Description | Store manager add a project to a course |
| Primary Actor | Logged in Store manager |
| Entry Condition | The course exists within the system |
| Exit Condition | The new project is added |
| Flow of event | 1. Store manager choose add project 2. System displays add project interface 3. Store manager choose a course 4. Store manager adds project name 5. System adds new project |
| Special Requirements | None |

|  |  |
| --- | --- |
| Title | Add a holiday |
| Description | Store manager adds a holiday |
| Primary Actor | Logged in Store manager |
| Entry Condition | None |
| Exit Condition | Holiday is added and students expiration date is adjusted |
| Flow of event | Store manager enters the holiday date and name  System add new holiday date, and scan through all students course progress, extend their expiration date if |
| Special Requirements | None |