**D:\Work\RoboOfficeNXT\Artwork\RoboOfficeNXT.pngRoboOfficeNXT ERP System Analysis Report**

**System Analysis Overview**

Purpose of the RoboOfficeNXT system is to actively manage Robokids Enterprise infrastructure resources, across multiple locations, multiple users, and concurrent terminals. It automates most of the tasks that are currently done by the receptionist, instructors, and school managers.

**Architecture Components**

**Back-end database**: MySQL database server, installed in office desktop computer at each location. Accessible on the internet or LAN

**Client-side applications**: J2SE Desktop Application for office use. Java web-app solution for instructors in each room or students at home

**Non-Functional Requirements:**

**Memory Management**: Java Desktop Application requires JVM, with certain memory size restrict on each application. The Swing package and JDBC can use significant amount of memory that can overflow the JVM. Each run of the desktop application should not result in sluggish performance or memory leak.

**UI Intuitive**: Training time of each user of the system is expected to be less than 30 minutes. The user interface should be easy to use. The look and feel of the application should provide the user comfort visually.

**Fast start time**: The start time of the desktop Application and loading time of the web app should be less than 10 seconds. Optimally, 5 seconds.

**System daily backup**: The system should run a database backup daily, and store the backup file on the cloud or another server/terminal.

**Multiple locations**: the system is expected to distribute in more than one location, and database are expected to be accessible to each other. The headquarters should be able to view all location status.

**Multiple user groups**: the users of the system are divided into multiple user groups, and the system should support concurrent usability.

**Software Modules**

**Overview**

* User Control
* Human Resource/Staff Management
* Issues/Memo/Tasks Communication System
* System Tools/Logs/Utilities
* Administration / School Metadata
* Prospective Student/ Admission
* Current Student Management (Attendance, notices, timetable arrangement)
* Finance/Account Receivable

**User Control**

A user is generated when a personnel is created inside the system. A user has to belong to a user group, specified by the system. A user may be coupled with a staff within the system, therefore connecting real personnel with a system account.

By default the system has a “root” account, for initializing the system, edit user group permissions, and create school manager account. It can also see every user’s password in case of a user forget their password and question.

System User groups include

* Receptionist
* Instructor
* School Manager
* Accountant

Only School Manager can create or delete user accounts. However, School Manager cannot modify user group permissions. Only the “root” account can.

User cases

1. Login: A user type in username and password, and the system grants or denies the request. System logs the activity. System then displays controls and information that are only relevant to that user. All other access is limited.
2. Logout: A user clicks “logout” and the system exits for that user. System logs this activity. System return to login page.
3. Changing password and question: A user changes to new password, and select a question and answer.
4. Add/Delete/Edit User: A user can be deleted and added by Staff Management (See HR/Staff Management). User’s permission group can be edited.

**Human Resource/Staff Management**

Use Cases:

Work hours: Enter monthly availability, Enter hours actually worked, View Work Schedule for every month.

Payroll: Calculate staff monthly salary by hours actually worked, pay cheque.

Staff Management: Edit staff personal information, create new staff, and delete staff.

Scheduling: by a special user group, all staff’s work hour can be scheduled monthly

Holiday Policy: Set or delete holiday Policy, which day to take out of the system.

**Tasks/Memo Communication System**

This subsystem controls the inner-communication between each system users. (Refer to github issue system interface)

Tasks: Messages Created from a user to multiple users.

Messages are seen publicly for every user.

Messages have status of open and closed. Closed messages can be reopened.

Memo: Broadcasts for every user to see

Read or unread status

User Cases:

Pop-up Windows: notify user if there is any new tasks/Memos after user’s last logout.

View all tasks: See all tasks

View my task: see all tasks received.

Created by me: View all tasks received

Create task & memo: Every user can create task and memo. For task, recipient(s) need to be specified. Sender can check the status of the task, see if it’s done.

Close Tasks: Once a recipient completes a task, its status can be changed to “closed”.

Delete memos: School Manager can delete only Memos.

**System Tools/Logs/Utilities**

This subsystem concerns with the activities of the system itself. Functions including daily database backups, system activity logs etc.

Daily database backups: Database is expected to be backed up daily. Headquarters should have all copies of the backup in every location. If required, headquarters should be able to view/manipulate the backup copies of the other locations.

System logs: records all important activities done on the system.

Automated Emails for school events: student absent, makeup class required, birthday notice, special promotion, etc.

Receptionist will be able to view automated email sent.

**Administration/School Metadata**

School information such as classroom, courses, timeslots, payment methods are expected to be modified through this module. Only School Manager has privilege to this module (initially)

Use cases:

Courses information: name, duration, price, pay period, description

Classroom information: Capacity, location, time available

Timeslots information: day of week, duration, start time.

Material information: NXT sets, etc.

Tax option

Anything that is the school configuration that is related to the ERP system

**Prospective Student/ Admission**

This module involves new students and enquiry students.

Use case:

Student information input: These are contact information regarding a prospective student, potential customer, enquiry students

Student Assessment: Evaluate a new member’s skill aspects

Student registration: Record student information, then direct to payment for paying for courses/timeslot selected.

**Current Students Management (Attendance, notices, timetable arrangement)**

This module manages all current active/inactive students.

Include all school information of a student.

Personal information

Class records history

Payment history

Student comments and progress rating report

Use cases

Attendance:

* View Weekly Schedule by Calendar.
* Print daily Attendance time table via Excel sheet
* Adjust attendance table for time conflict, manipulate student slot
* Record attendance for each day

Student activity

* Edit student personal information
* Change a student’s level or time slot
* Mark sick-notice absent notice for a student
* Enter Student progress report/rating by class records.
* Change a single class record information

Overview / Statistics

* Student statistics include active students, inactive students, enquiry students, archived students.
* Overdue tuitions, expired courses, need-to-pay soon students.
* Course finishing on time percentage. Age gender distribution etc.
* Class records history and payment history for a given student

**Finance/Account Receivable**

This module includes all financial aspect of the school. Including overall AR list, Archived payment history, Payroll information

Use Case:

New Payment: Receiving a new Payment for a student. Each payment can include more than one course and timeslot. The system needs to perform capacity check and course eligibility check.

Account Receivable: Statistic overview of all Payments made by all students, in chronological order.

Payment Fee List:

All payments have options:

1. Tax
2. Discount

Courses:

1. Pay per month. Pay [rate] for number of weeks a month, expire at the end of Month.
2. Pay by sessions. Pay [rate] \* [number of weeks], expire at the end of number of weeks.

Memberships:

1. Fixed start and finish date, and rate for each month.
2. Pay anytime, pay [rate/month] \* [finish date – current date].

Admin Fees (charge slot change etc) Material (NXT, LEGO sets ) Events (Competitions, entry fees, special FLL training fees):

1. Pay fixed rate.

Event : expire date.

Payment Methods: Cash, Debit, Credit, Cheque. Effect?

Courses:

Start time + classroom + courses = 1 Class section.

Ex, Monday 16:00 + 2A classroom + 2A course = 周一下午2A课程。

Every student can have multiple courses.

Every progress: available credit, expire date,