We are already assuming that the system you have support multi schools. Kindly correct us if we are wrong.

Payment procedures:

The course fees collection could be set based on one of the following two methods:

1) Standard:

   Standard method is Straight forward. It checks the number of hours a student studied monthly and charges him for that till all he finishes all the course hours required and pay all the fees.

   Payment due one month later for each month units. For example, if a student studies 20 hours in September and we charge SEK 10 per hour (SEK is the Swedish currency), then the student fee for September is 20 hours * SEK 10 = SEK 200 and it is due one month later, End of October.

2) Dynamic:

The dynamic payment procedure has one default node (On Test completion-successfully), and payment segments. The admin set the number of segments and its settings as shown in the following two samples:

Dynamic Payment procedure sample 1:

Segment 1 (Name):

Payment sum due: /20/ % of course price

Due date: in /30/ days (Due date for dynamic segments is always set From student_enrollment_date onwards)

Segment 2 (Name):

Payment sum due: /30/ % of course price

Due date: in /60/ days (Again, Due date for dynamic segments is always set From student_enrollment_date onwards)

On Test completion-successfully (This is the default one)

Exam: ¤ Passed (Teacher ticks the box here when the student passes the test successfully and that triggers the due date

Payment sum due: /50/ % of course price

Due date: in /20/ days (Only this due date is set from the day the student passes the exam, not from student enrollment date)

Dynamic Payment procedure sample 2:

Segment 1 (Name):

Payment sum due: /10/ % of course price

Due date: in /20/ days (Due date for dynamic segments is always set From student_enrollment_date onwards)

Segment 2 (Name):

Payment sum due: /30/ % of course price

Due date: in /40/ days (Again, Due date for dynamic segments is always set From student_enrollment_date onwards)

Segment 3 (Name):

Payment sum due: /40/ % of course price

Due date: in /60/ days (Again, Due date for dynamic segments is always set From student_enrollment_date onwards)

On Test completion-successfully (This is the default one)

Exam: ¤ Passed (Teacher ticks the box here when the student passes the test successfully and that triggers the due date

Payment sum due: /20/ % of course price

Due date: in /20/ days (Only this due date is set from the day the student passes the exam, not from student enrollment date)

Number of periods is variable so as the values between /…/ in the above structure.

Then when I create a new course, I can choose the unit scaling of this specific course, I set Whether to apply standard payment procedure to collect fees for this course or a dynamic procedure from a dropdown list that shows procedures created already (as explained above).

Fees collection generated reports

To give you an idea first about the system in Sweden, it is not the student that pays, as this type of education in Sweden is paid by the municipalities to the schools from students' behalf. This can make it clearer why we need such reports and how it needs to be generated.

- The Standard-based procedure monthly report: A monthly report listing all the fees (by student then as a total sum) required for that month. The report will have a report ID, the month it targets, number of hours studied by each student, amount due by student, Total sum of hours taught for students in that month, total sum of fees to collect and payment due date.
- I can select any date and get a report of all the Standard-based payments due (the collection of "The Standard-based procedure monthly report" mentioned above, but not paid yet, which includes the Report id, report month, amount due, amount paid, amount left to collect, Due date (from the initial report).
- The dynamic based monthly report: A monthly report listing all the fees (by student then as a total sum) required for that month. The report will have a report ID, the month it targets, Student segment due, amount due by student, total sum of fees (for all students) to collect, and payment due date.

- I can select any date and get a report of all the dynamic-based payments due (the collection of "The dynamic-based procedure monthly report" mentioned above, but not paid yet, which includes the Report id, report month, amount due, amount paid, amount left to collect, Due date (from the initial report).

What happens is that we send this report (not through the system, but it will also be good if we can send it via email through the system) to the municipality and we get paid based on the total number of hours delivered to students per month. They pay us one month later. For example, the report of September is paid end of October, but sometimes they can delay the payment or forget it (You know public sector sucks everywhere, even in Sweden!).

In the system we also need a page where we find reports list (Report id, name, date issued, date due, date paid, and a tick for paid/unpaid.

Other modifications to the current system

Additional course fields on course creation:

- Course fee collection procedure (dropdown list showing the procedures created based on what We explained above in the document)
- Course # of hours (required if the collection procedure is set to standard)
- Number of default hours recorded for student per one attended session (Let's say the student attended the course on Monday and the course in Monday lasts for two hours, then we need to add 2 hours to the number of hours this student completed this month as this is important for fees collection for this month).

Course content (I think you have that as syllables in the system):

Teacher can add a pdf (or many) to each lesson where students can see them (1 or many) and student see the pdf as flipping book (1 page at a time to be swiped).

Another Assignments type to add to current system assignments:

Teacher can add assignment (1 or many), each assignment has a deadline, and teacher can send it to all students in this class/lesson, or specific student. Each assignment can have the questions as a pdf that opens in the page and has a text box for the student to write the answers. Student get notified of the new assignment. When students go to the assignment, he writes the answers in the assignment text box and send it.

Teach can see all assignment answers, put the grade and can send the right answers to all students taking this assignment.

New fields for Student profile:

- Student personal number: 12 digits number added manually – and it is unique.
- Student nationalities (many) and languages (many)
- Student education background
- Gender in Sweden should include 3 types (by law): Male, Female, other.
- Start date: default value is the day the profile is created. It identifies when a student gets enrolled into a course (This field should be required as it is important for the dynamic procedure fee collection as explained above).
- End Date (both Start date and end date are editable),
- Test Date (editable)
- Teacher test Reminder: (XXX days before test date, where a notification -within the system and via an email- is sent to school admin and teacher to make sure the test is prepared on time.
- Exam result (Pass, Fail) as set by the teacher. Note: Once the exam status become passed, it triggers the relevant payment in the dynamic fee collection procedure as explained above.

Student status

The student added to the system has to be added to one of the two mentioned programs and will have one of the following statuses:

- Inactive: A student added to the system but yet to be enrolled into a program-course.
- Active: A student who is studying a course.
- Pending: A student who finished the course but didn't pass the exam yet.
- Incomplete: A student who was withdrawn from the course at a certain date.

We should be able to see a report for Students-status and filter it by status type and/or by the course the student is enrolled in.

Study plan:

There will be a template (we will send you that template on word so you convert it to form). Teacher fills a study plan for every student and both can see it. Teacher can save a draft for a Study plan to use it for other students later. Teacher apply one Study plan for all class/batch students and in this case the same study plan goes to all students with required changes like name, personal number, etc... being dealt with thru the system automatically.

Progress bar:

Shown to student, school admin & teacher. Progress bar shows the progress of a student in his course how far he finished and what is left.

Attendance module:

I like the way you setup your attendance module, but we will need to add a value by hour for each of the listed statuses you have there:

Present: Means that the student attended the whole session, so we add the “Number of default hours recorded for student per one attended session” value to the number of hours attended by this student this month (for standard procedure fee collection). Check the explanation of: “Number of default hours recorded for student per one attended session” in the **Additional course fields on course creation** paragraph explained above.

Custom: We create a new status called “Custom” where we input the exact number of hours a student attended in the session (in this case he didn’t attend the whole session time).

For Absent and holiday statuses: we add zero hours for the number of hours attended by the student for this month.

Taking attendance by teacher is required. If the teacher misses taking an attendance, he gets notified about that. Every morning, the system sends a report for the admin about all the sessions where attendance was not taken the day before.

Calendar:

Where user can see the timings of his courses, assignments and tests

Interface:

We will discuss that after you analyze the customizations required.