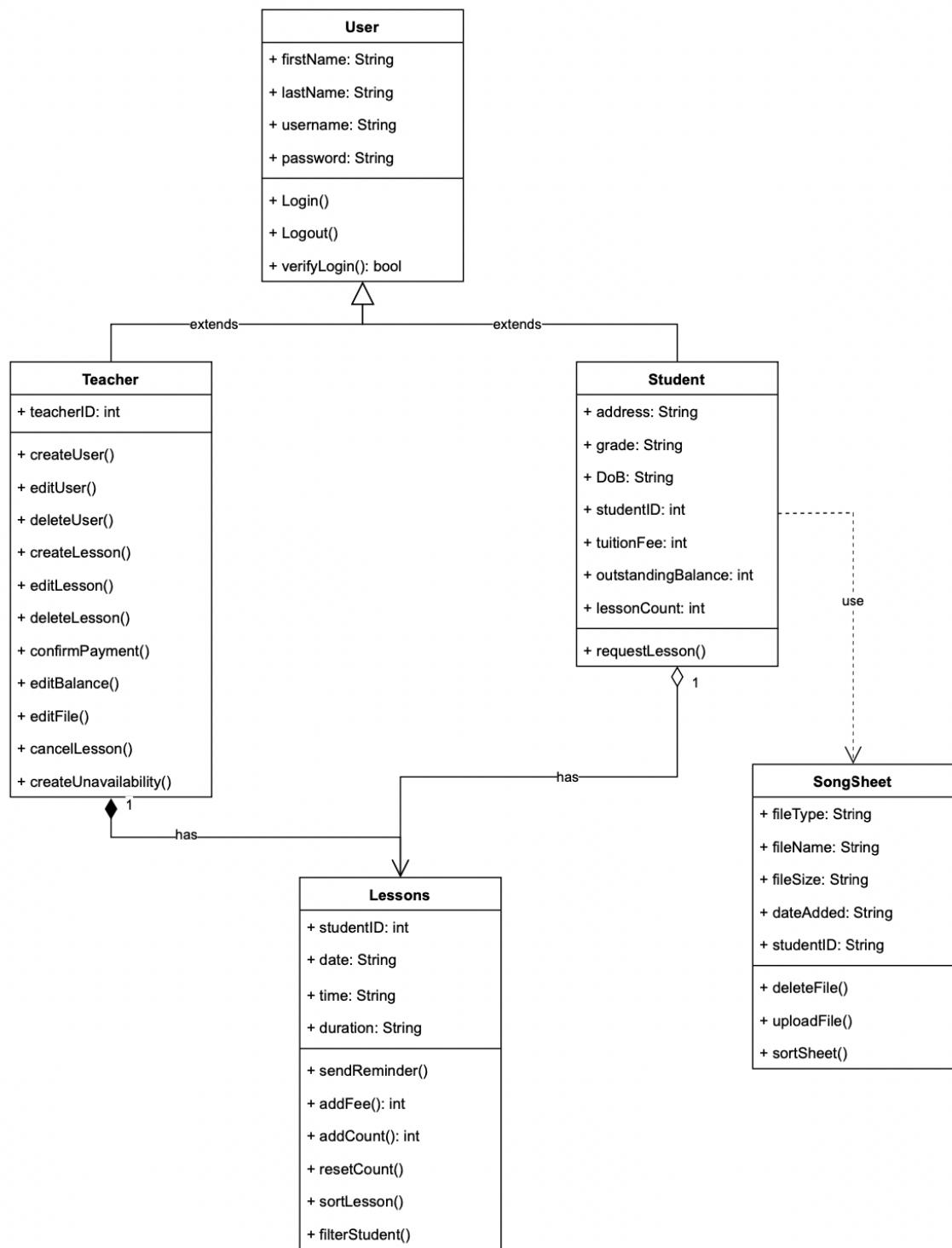


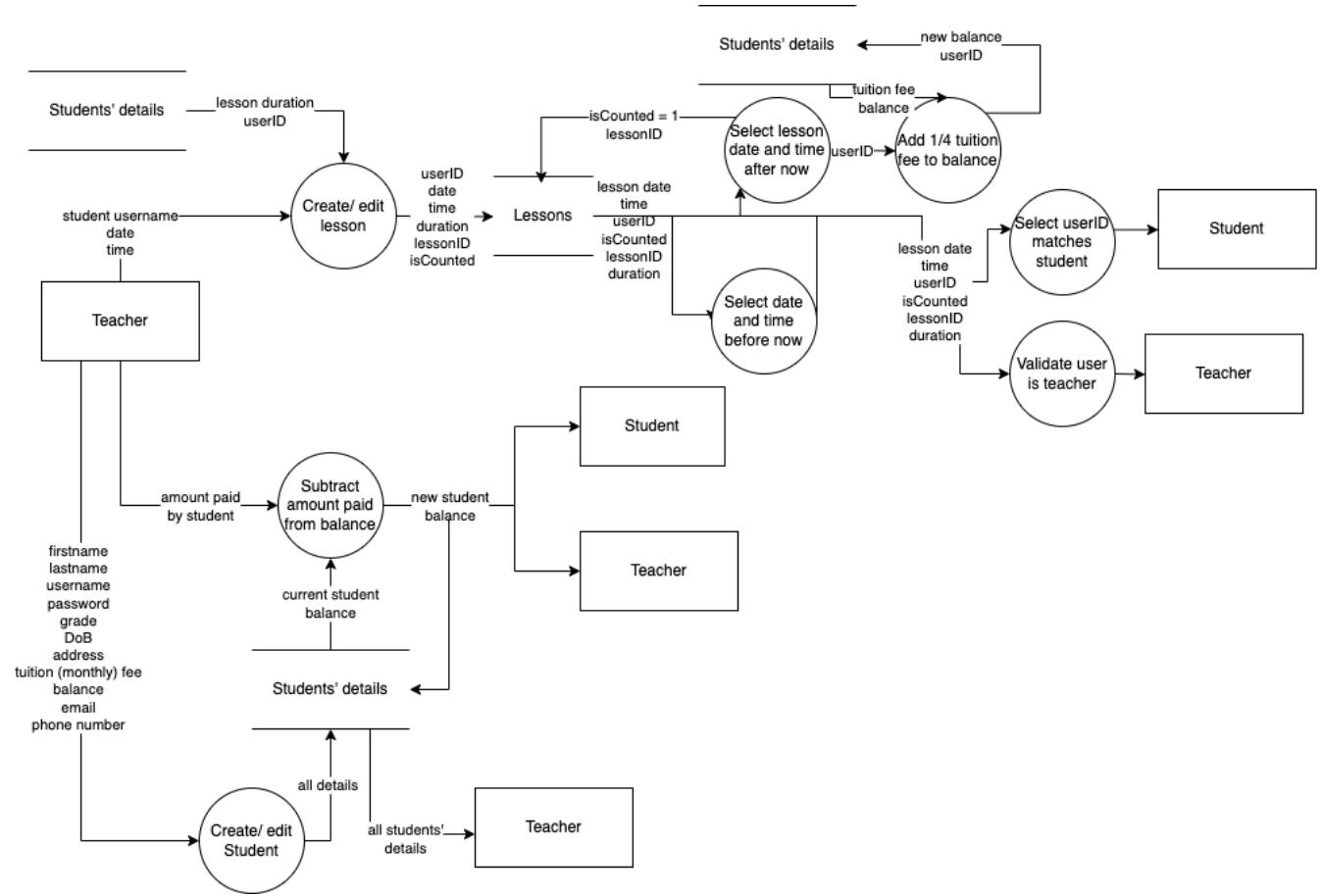
Test Plan

Nature and Rationale of Test	Example
- Payments calculated properly - Ensure lesson Count is reset once it reaches 4	- Make tuition fee \$200, \$0, \$-5 - Input 8 lessons - Balance is \$400, \$0, ERROR
- Check program outputs correct student and lesson details - Ensure client can make typos and still see the desired student	- Input first name with one typo - Output should show the student with their lesson
- Ensure chronological order is correct	- Input multiple lessons with different days and times - Ensure lessons are displayed in descending order
- Ensure notifications send to client	- Input booking request - Output notification
- Ease of finding song sheets	- Input word - Output song sheet with name including that word
- Ensure balance tracker is functional	- Input amount paid, \$20 - Output new balance, which is old balance minus amount paid, \$50 → \$30
- Make requesting/ creating new lessons faster	- Input date - Output lessons on that date
- Disallow overlapping lessons	- Input different lessons: - lesson starts slightly earlier than other and finishes during other - start and end within other lesson - start and end before and after other lesson - Should not be allowed, other combinations should be allowed
- Ensure student with duplicate username does not exist	- Create a new user with a username that already exists ('aname') - Should not be allowed - Create a new user with unique username ('newusername123') - Should be allowed

UML

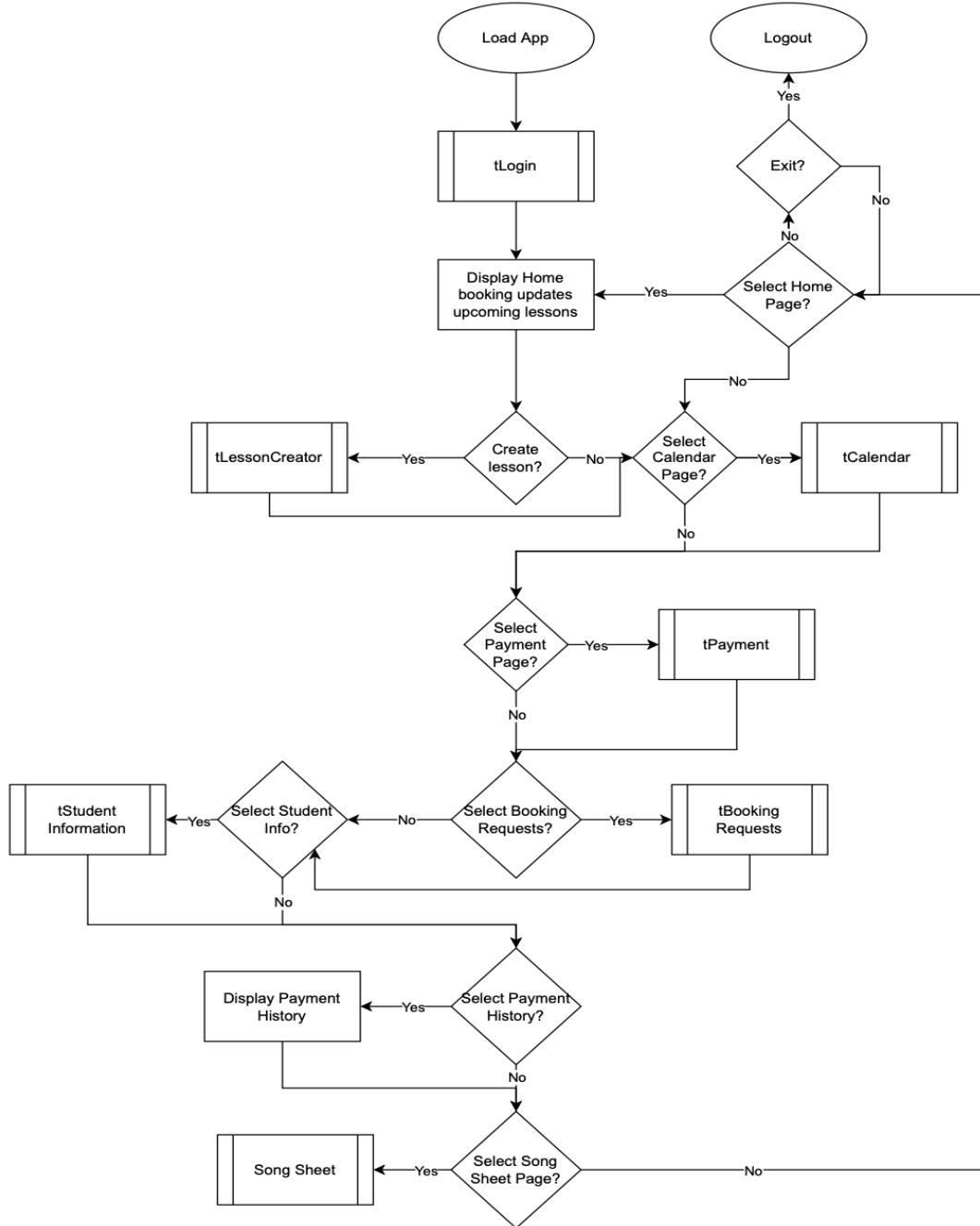


Data Flow Diagram

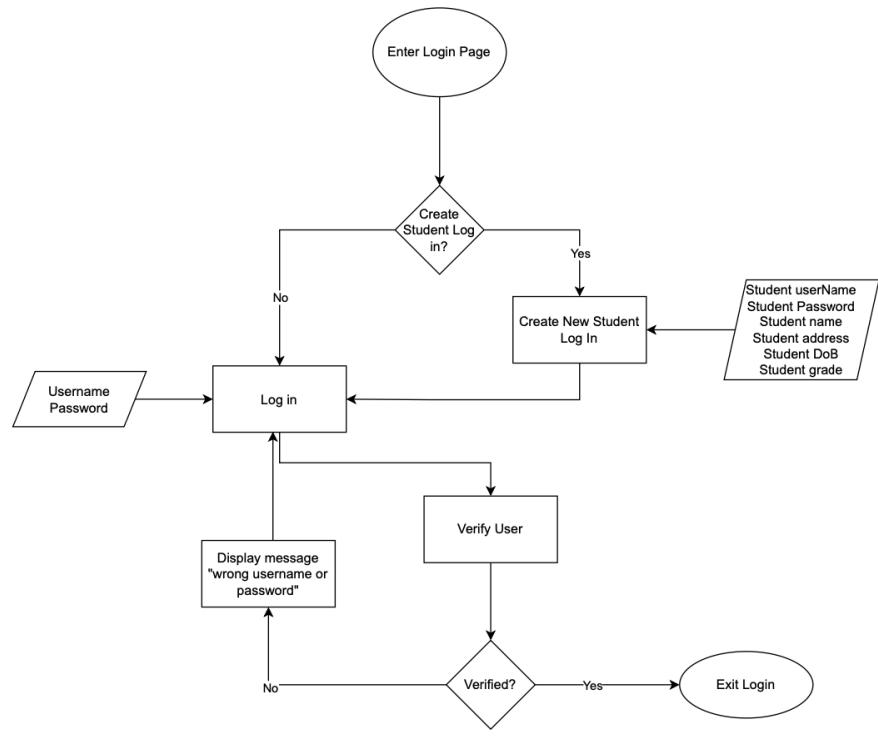


Flowcharts - Teacher

Flowchart 1. teacherApp



Flowchart 2. teacherLogin



Log in:

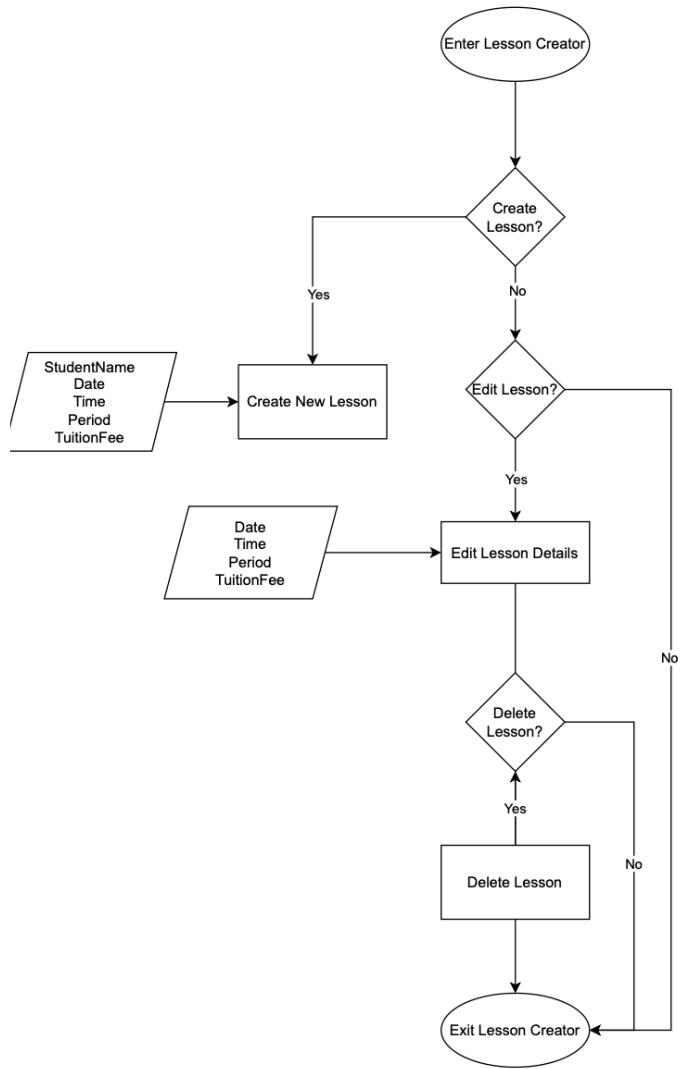
```

// get user with username and password from DAO
user USER = login(USERNAME, PASSWORD)
// check username and password exists for one user in DAO
if (USER != null)
    int INDEX = 0
    ArrayList<user> USERLIST = DAO doGetUserList()
    int LESSONCOUNT;
    loop INDEX from 0 to USERLIST.size()
        int ID = USERLIST.get(INDEX).getUserID()
        ArrayList<lesson> USERLESSONS = DAO.findUserLessons(ID)
        int INDEX2 = 0
        loop INDEX2 from 0 to USERLESSONS.size()
            Date DATE =
                USERLESSONS.get(INDEX2).getLessonDate()
            Time TIME =
                USERLESSONS.get(INDEX2).getLessonTime()
            int LID = USERLESSONS.get(INDEX2).getLessonID()
            LocalDateTime DT = LocalDateTime.parse(DATE+TIME)
            LocalDateTime NOW = LocalDateTime.now()
            if (NOW.isAfter(DT))
                DAO.setIsCounted(1)
                LESSONCOUNT = DAO.getLessonCount() + 1
            end if
    end loop

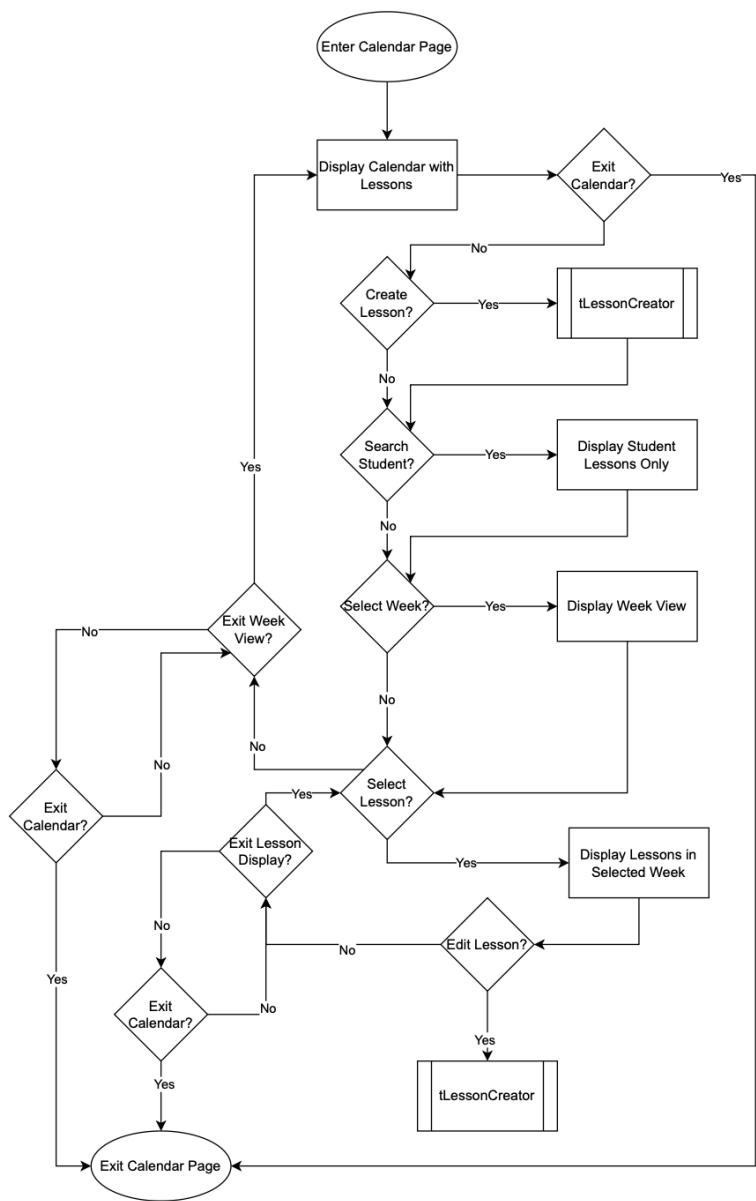
```

```
end loop
loop while LESSONCOUNT > 4
    int TUITIONFEE = DAO.getTuitionFee(ID)
    int BALANCE = DAO.getBalance(ID)
    BALANCE = BALANCE + TUITIONFEE
    DAO.setBalance(BALANCE)
    LESSONCOUNT = LESSONCOUNT - 4
end loop
DAO.setLessonCount;
if (user.isTeacher == true)
    return "homepageTeacher"
else if (user.isTeacher == false)
    return "homepageStudent"
end if
end if
msg = "username or password is wrong"
return "index"
```

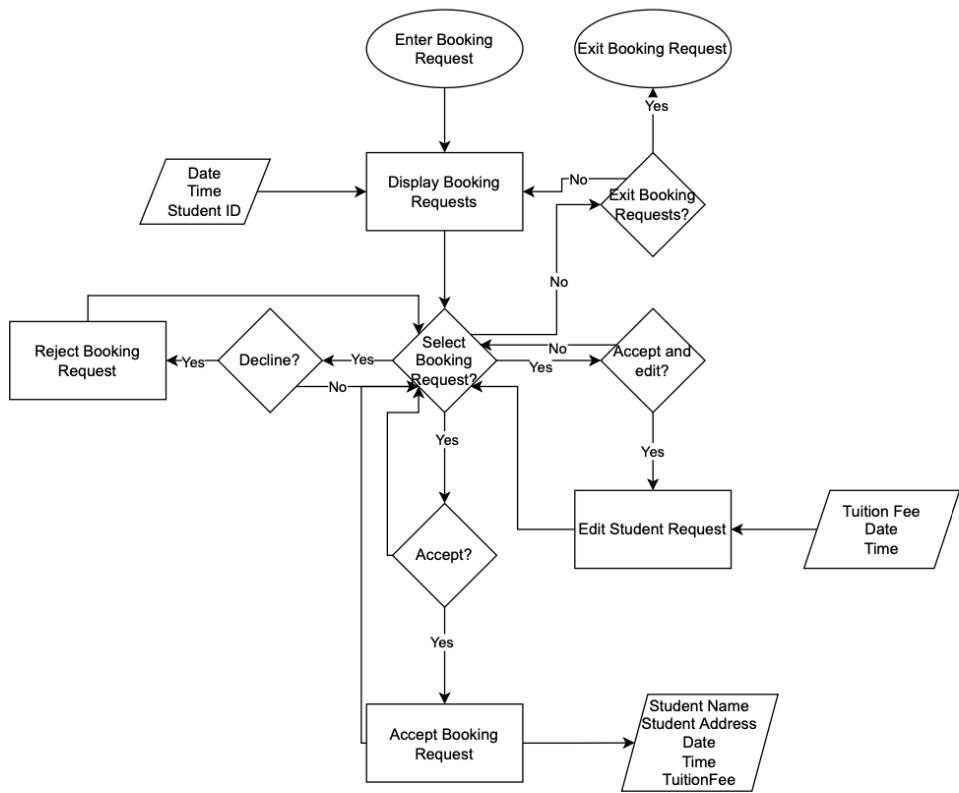
Flowchart 3. teacherLessonCreator



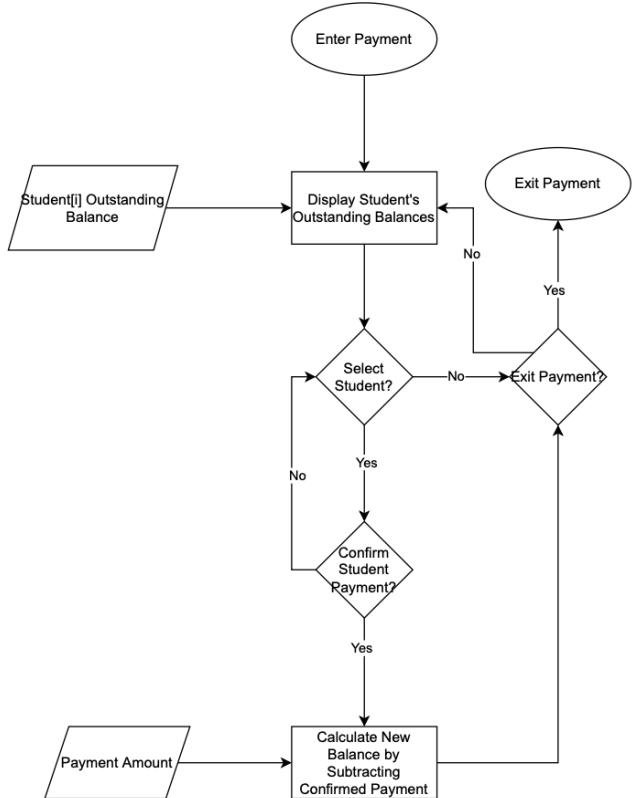
Flowchart 4. teacherCalendar



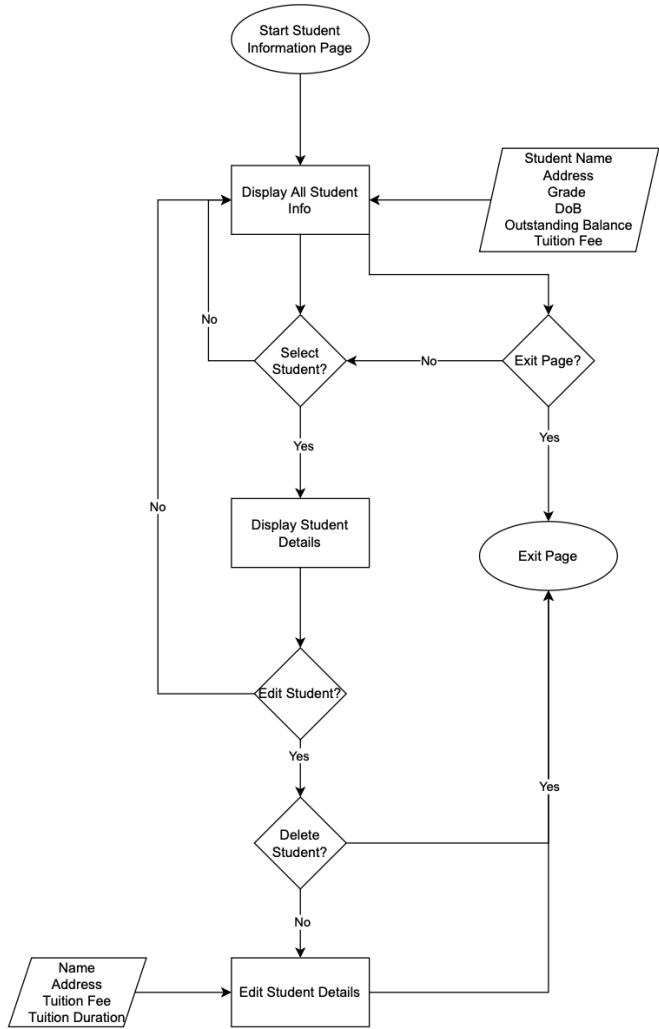
Flowchart 5. teacherBookingRequest



Flowchart 6. teacherPayment

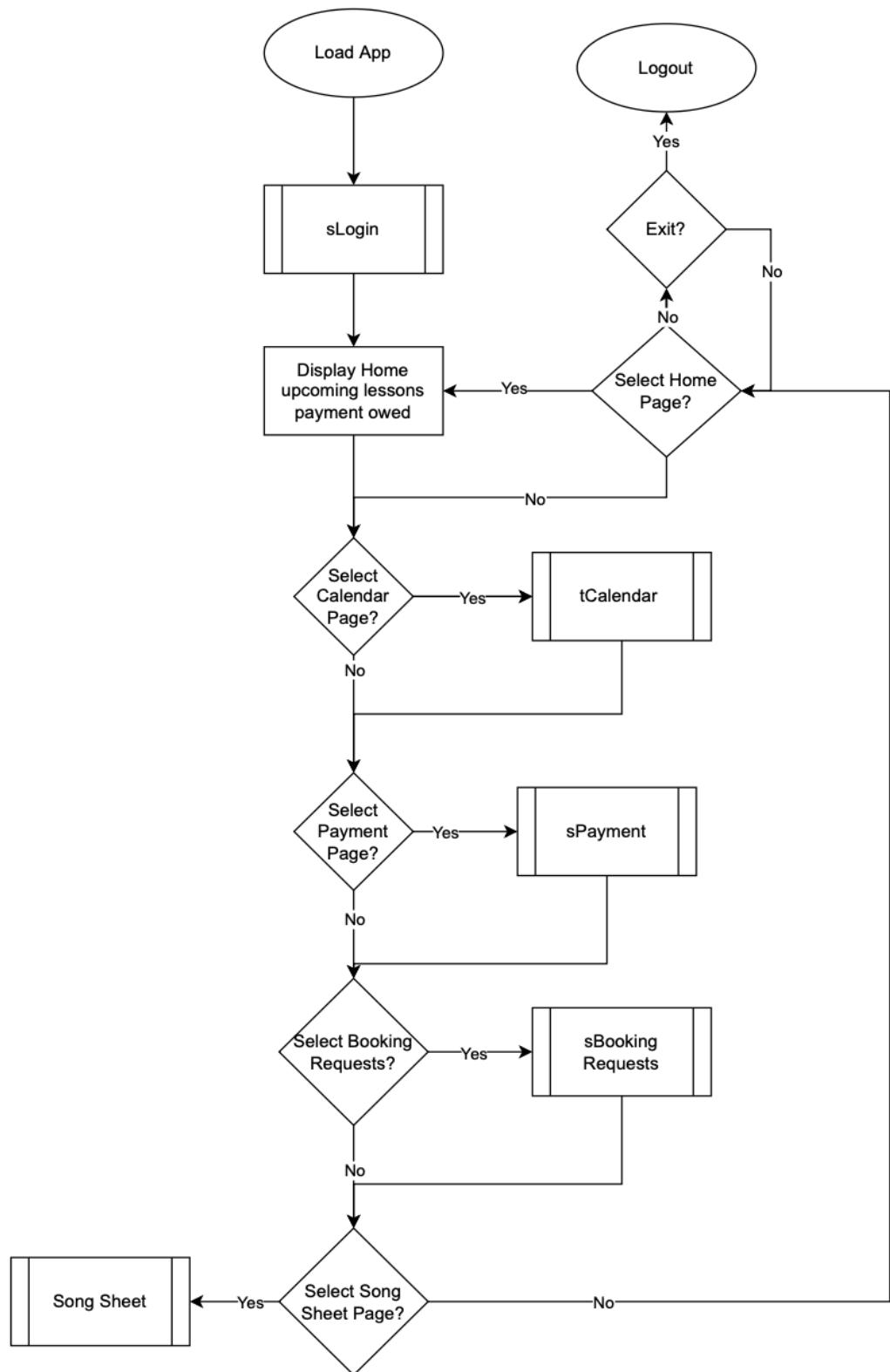


Flowchart 7. teacherStudentInfo

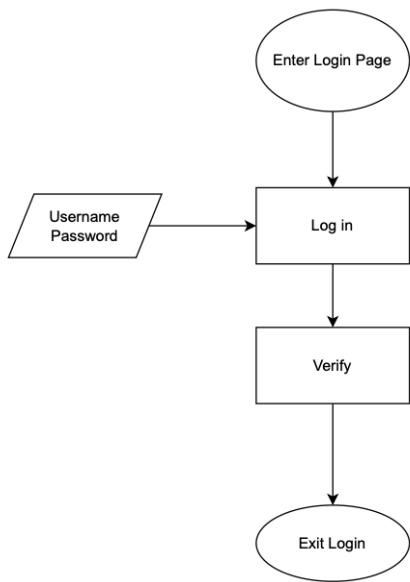


Student

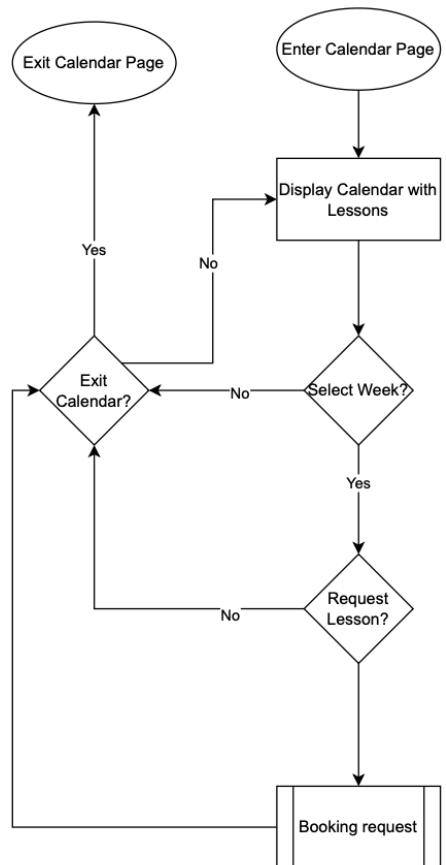
Flowchart 8. studentWebsite



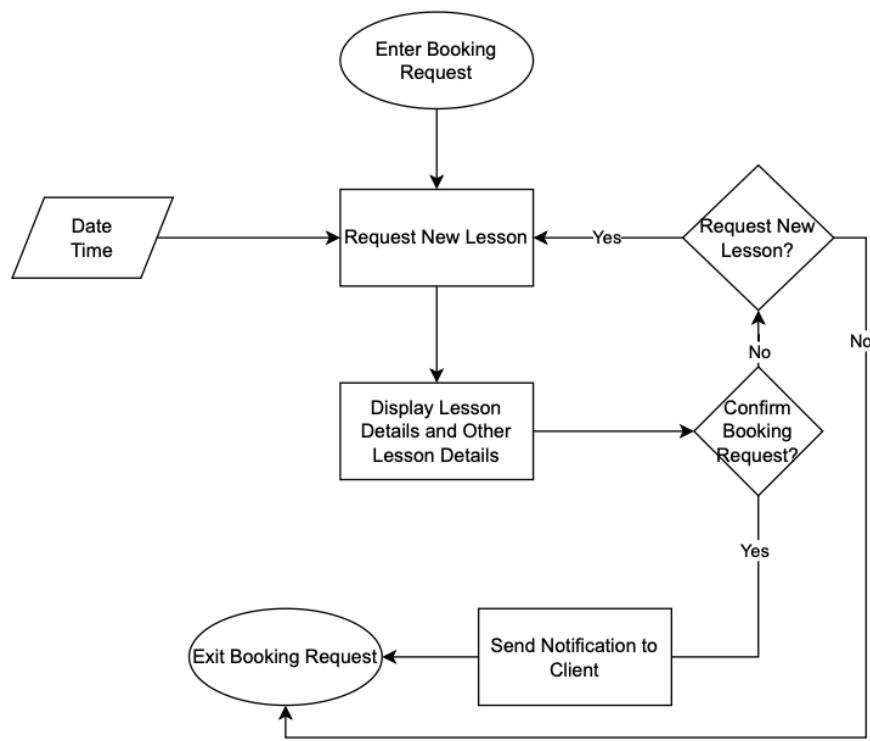
Flowchart 9. studentLogin



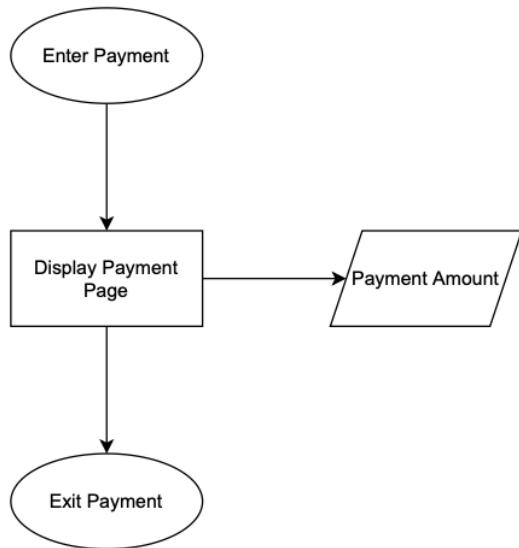
Flowchart 10. studentCalendar



Flowchart 11. studentBookingRequest

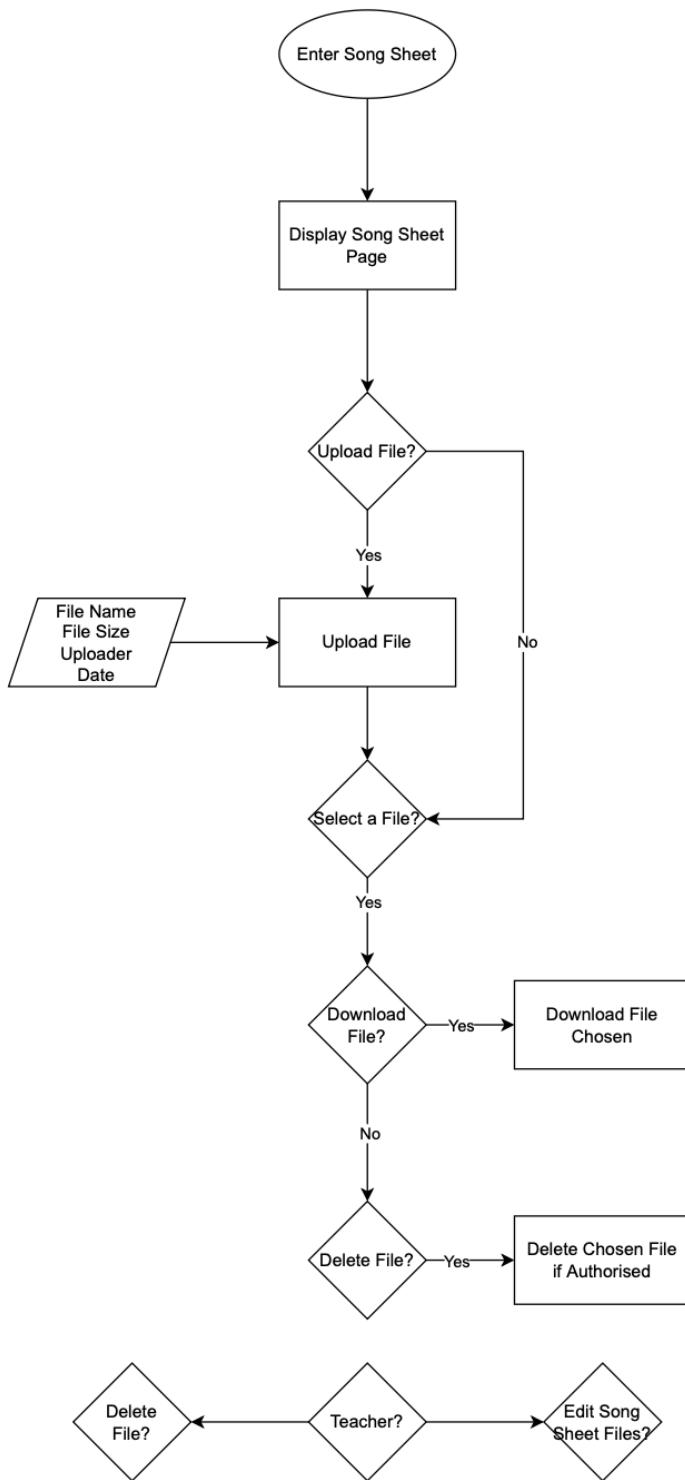


Flowchart 12. studentPayment

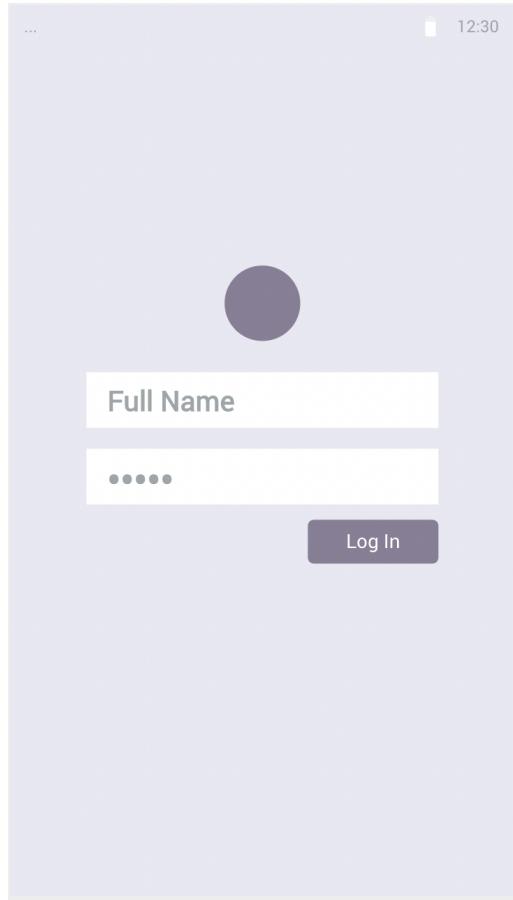


All Users

Flowchart 13. songSheet



Mock GUI for Teacher Application



Login Page

Purple chosen due to client's preference

A screenshot of a mobile application's home screen. At the top left is a "Home" button and a three-line menu icon. Below is a grid of booking requests. The first column is labeled "Today" and the second column is labeled "Tomorrow". Each request card contains a user icon, name, address, and time range. A red badge at the top left of the grid indicates "3 New Booking Requests".

Today	Tomorrow
FirstName LastName Address 6:00 - 7:00	FirstName LastName Address 6:00 - 7:00
FirstName LastName Address 7:30 - 8:30	FirstName LastName Address 6:00 - 7:00
FirstName LastName Address 10:00 - 11:00	FirstName LastName Address 6:00 - 7:00
FirstName LastName Address 2:00 - 3:00	FirstName LastName Address 6:00 - 7:00
FirstName LastName Address 4:00 - 5:00	FirstName LastName Address 6:00 - 7:00
FirstName LastName Address 6:00 - 7:00	FirstName LastName Address 6:00 - 7:00
FirstName LastName Address 6:00 - 7:00	FirstName LastName Address 6:00 - 7:00
FirstName LastName Address 6:00 - 7:00	FirstName LastName Address 6:00 - 7:00

Home Page

Calendar

September

Create New Lesson

search...

M T W T F S S

12:30

Calendar Page

Client can select a week by clicking on it at any point of its horizontal stretch, which navigates to the **Week View Page**

Week 1

September

Create New Lesson

search...

M 1 T 2 W 3

T 4 F 5 S 6 S 7

12:30

Week View Page

12:30

Day

September

search...

Create New Lesson

M 1 T 2

FirstName LastName
Address
6:00 - 7:00

Day View Page

Detailed view of client's chosen day and following day

12:30

Lesson Editor

Selected Lesson Select Student/ Create New

Student Name Student Name

Date DD MM YYYY

Lesson Duration 00:00

Start 00:00 PM End 00:00 PM

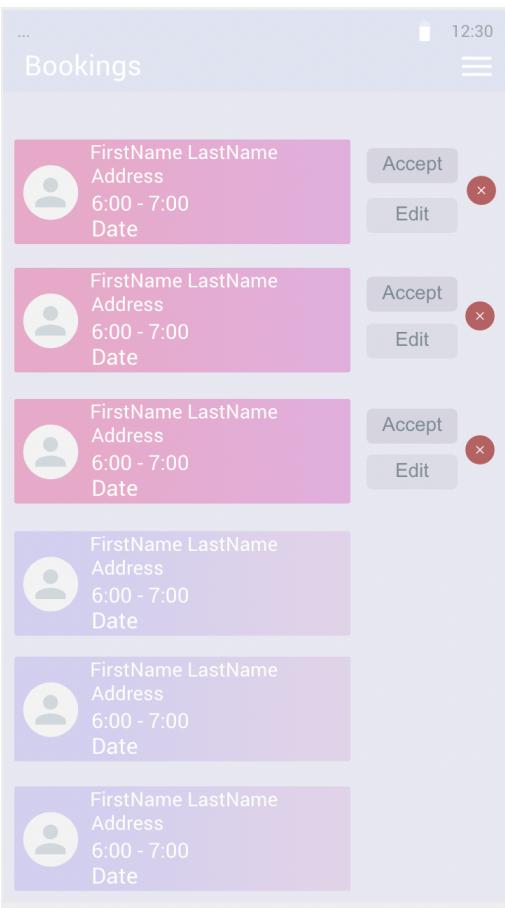
Fee \$ XXX / lesson

Address Student Name

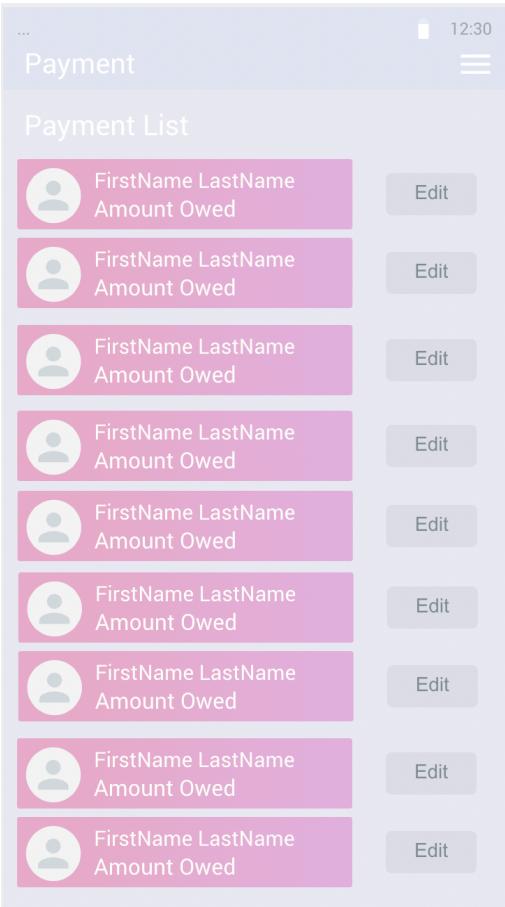
Contact (XX) XXXX-XXXX

Delete Cancel Save

Lesson Editor



Booking Request Page



Payment Page

12:30

Payment Editor

Selected Student: Select Student

Student Name: Student Name

Balance: \$ XXX

Payment Confirmation: \$ XXX

New Balance: \$ XXX

Cancel **Save**

This screenshot shows a mobile application interface titled "Payment Editor". At the top, it displays the time as 12:30. Below the title, there is a dropdown menu labeled "Selected Student" with the placeholder "Select Student". Underneath this is a text input field for "Student Name". To the right of the student name field are two input fields: one for "Balance" showing "\$ XXX" and another for "Payment Confirmation" also showing "\$ XXX". Below these is a field for "New Balance" with the value "\$ XXX". At the bottom of the screen are two buttons: a light gray "Cancel" button and a dark gray "Save" button.

Payment Editor

12:30

Student Info

Create New Student

FirstName LastName	Address	Year	Tuition Fee	View	Edit
FirstName LastName	Address	Year	Tuition Fee	View	Edit
FirstName LastName	Address	Year	Tuition Fee	View	Edit
FirstName LastName	Address	Year	Tuition Fee	View	Edit
FirstName LastName	Address	Year	Tuition Fee	View	Edit
FirstName LastName	Address	Year	Tuition Fee	View	Edit
FirstName LastName	Address	Year	Tuition Fee	View	Edit

This screenshot shows a mobile application interface titled "Student Info". At the top, it displays the time as 12:30. Below the title, there is a button labeled "Create New Student". The main area contains a table with six rows, each representing a student record. Each row includes a profile icon, first and last names, address, year, tuition fee, and "View" and "Edit" buttons. The table has a light purple header row and white data rows.

Student Information Page

12:30

Student Viewer

Selected Student

Student Name

DoB Age

Year

Address

Block No Unit No

Lesson Duration

Fee / lesson

Contact

Student Viewer

12:30

Student Editor

Selected Student

Student Name

DoB Age

Year

Address

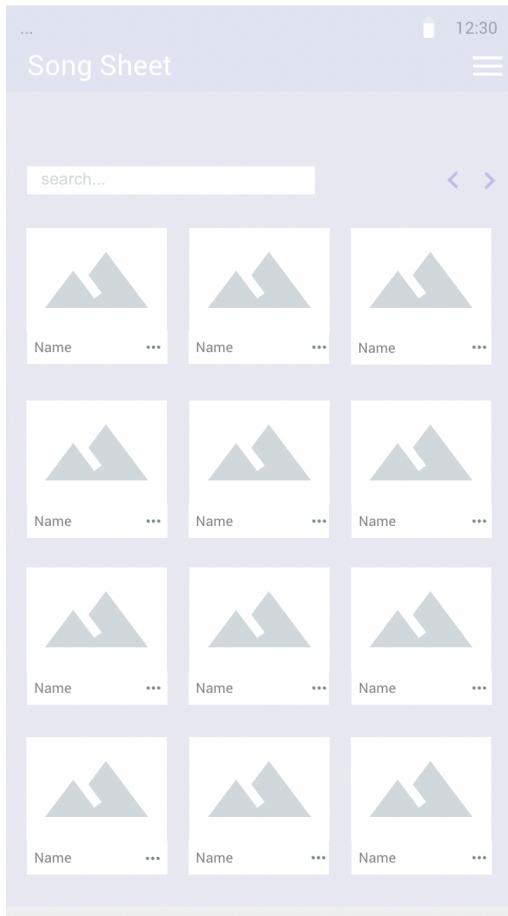
Block No Unit No

Lesson Duration

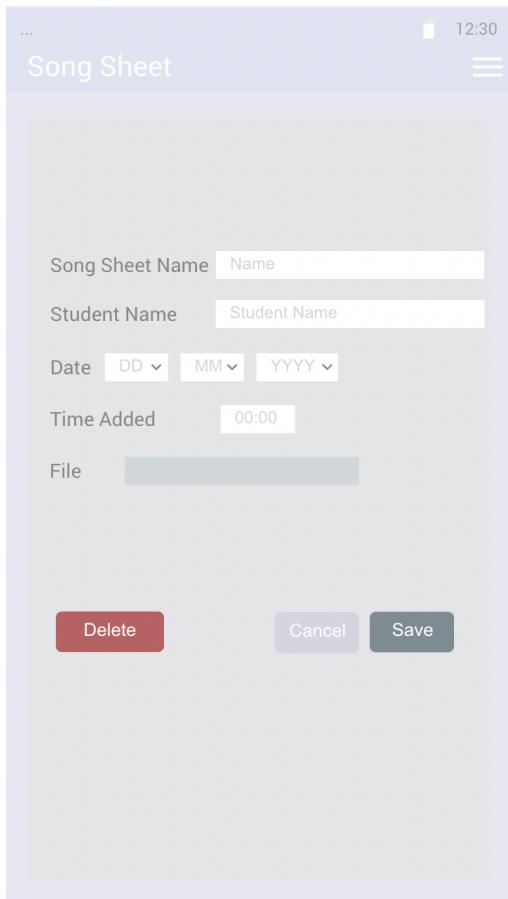
Fee / lesson

Contact

Student Editor



Song Sheet



Song Sheet Editor