Story

Teachers are in a hurry to get from one course to another in-between a few minutes. On these ways they have to manage having the lists, books and papers needed for the next lesson.

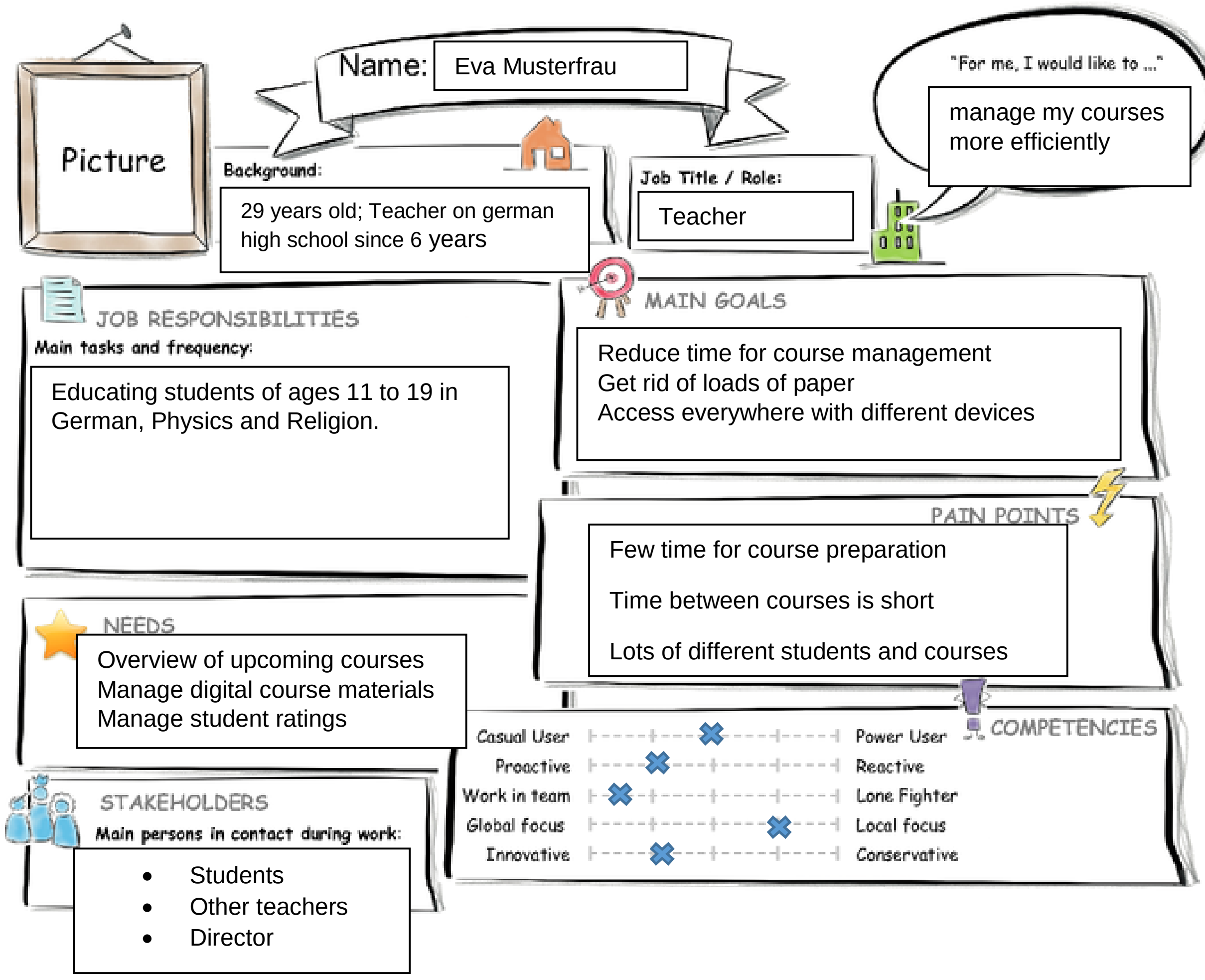
Knowing where to go, which students wait in the next lesson with their individual problems and what stuff to teach next, is a daily challenge in school.

Plans for the lessons are often made weeks or days before the courses take part and the teacher has to remember the progress of the lessons. The papers which have been designed long before have to be found and to be brought into the classes.

To organize the teacher's everyday business an App could help. The timetable shown on a mobile device should tell the teacher where to go next. The management of the students' personal data including pictures to remember their names, the grades of exams and lessons before and information about former lessons and circumstances of the classes could be taken on a mobile device and it wouldn't be necessary to have lists and calendar in paper form.

Also it could be really helpful to access papers written at home, from a mobile device in school, so the teacher does not have to search for copies given out before or take his textbook.

Persona



Picture

Name: Eva Musterfrau

"For me, I would like to ..."
manage my courses more efficiently

Background:
29 years old; Teacher on german high school since 6 years

Job Title / Role:
Teacher

JOB RESPONSIBILITIES
Main tasks and frequency:
Educating students of ages 11 to 19 in German, Physics and Religion.

MAIN GOALS
Reduce time for course management
Get rid of loads of paper
Access everywhere with different devices

PAIN POINTS
Few time for course preparation

Time between courses is short

Lots of different students and courses

NEEDS
Overview of upcoming courses
Manage digital course materials
Manage student ratings

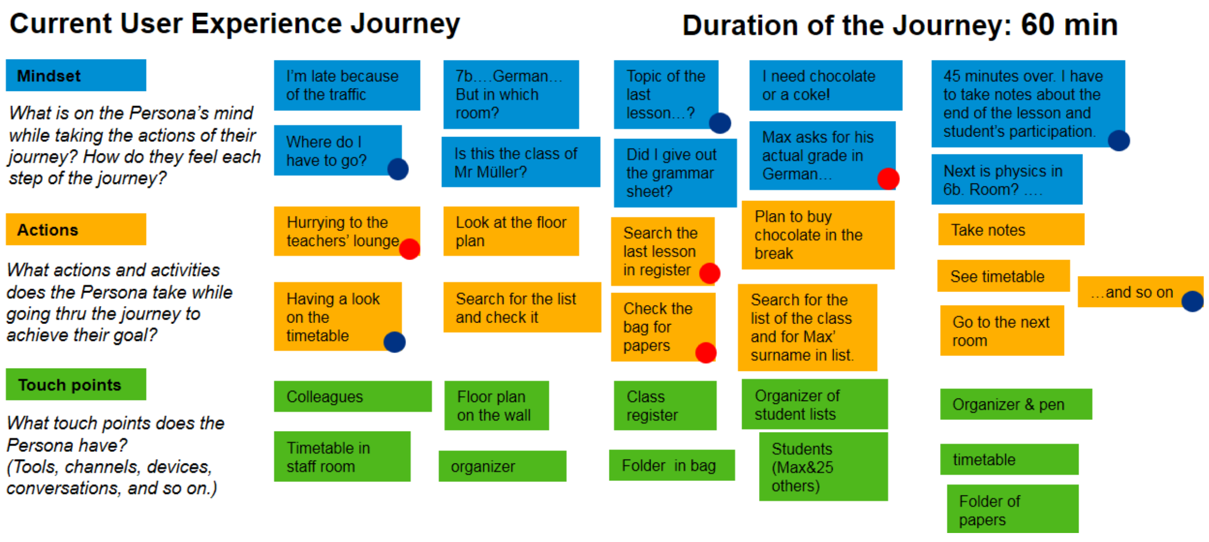
COMPETENCIES

Casual User			✖			Power User
Proactive		✖				Reactive
Work in team	✖					Lone Fighter
Global focus				✖		Local focus
Innovative		✖				Conservative

STAKEHOLDERS
Main persons in contact during work:

- Students
- Other teachers
- Director

User Experience Journey



Current User Experience Journey

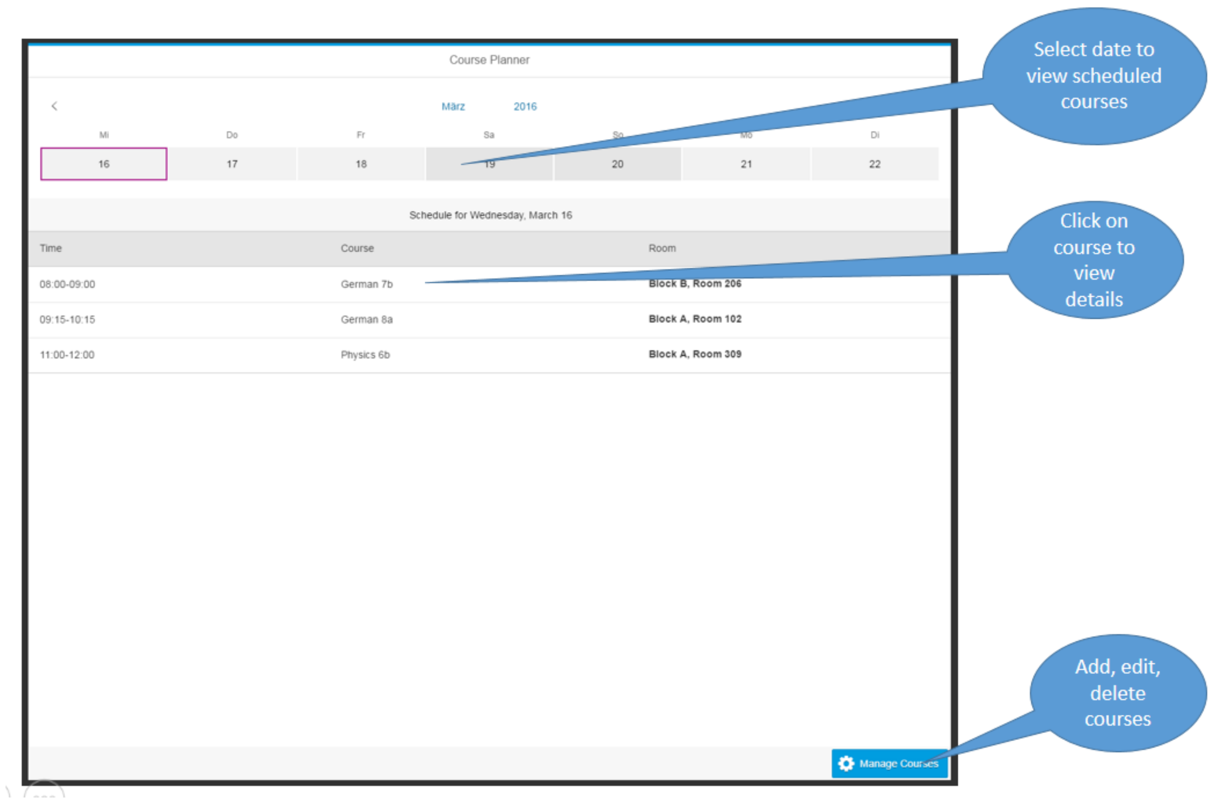
Duration of the Journey: 60 min

Mindset *What is on the Persona's mind while taking the actions of their journey? How do they feel each step of the journey?*	I'm late because of the traffic Where do I have to go?	7b….German… But in which room? Is this the class of Mr Müller?	Topic of the last lesson…? Did I give out the grammar sheet?	I need chocolate or a coke! Max asks for his actual grade in German…	45 minutes over. I have to take notes about the end of the lesson and student's participation. Next is physics in 6b. Room? ….
Actions *What actions and activities does the Persona take while going thru the journey to achieve their goal?*	Hurrying to the teachers' lounge Having a look on the timetable	Look at the floor plan Search for the list and check it	Search the last lesson in register Check the bag for papers	Plan to buy chocolate in the break Search for the list of the class and for Max' surname in list.	Take notes See timetable Go to the next room …and so on
Touch points *What touch points does the Persona have? (Tools, channels, devices, conversations, and so on.)*	Colleagues Timetable in staff room	Floor plan on the wall organizer	Class register Folder in bag	Organizer of student lists Students (Max&25 others)	Organizer & pen timetable Folder of papers

Point of View

Eva, the teacher, needs an overview of her upcoming courses and the rooms in which they take place. In the same moment she needs an efficient way to take notes about the last lesson and to grade the students' participation. She also needs a way to manage her course materials digitally, so that she doesn´t have to spend so many time on organization in order to concentrate on her main job of educating students.

Mockup


Course Planner

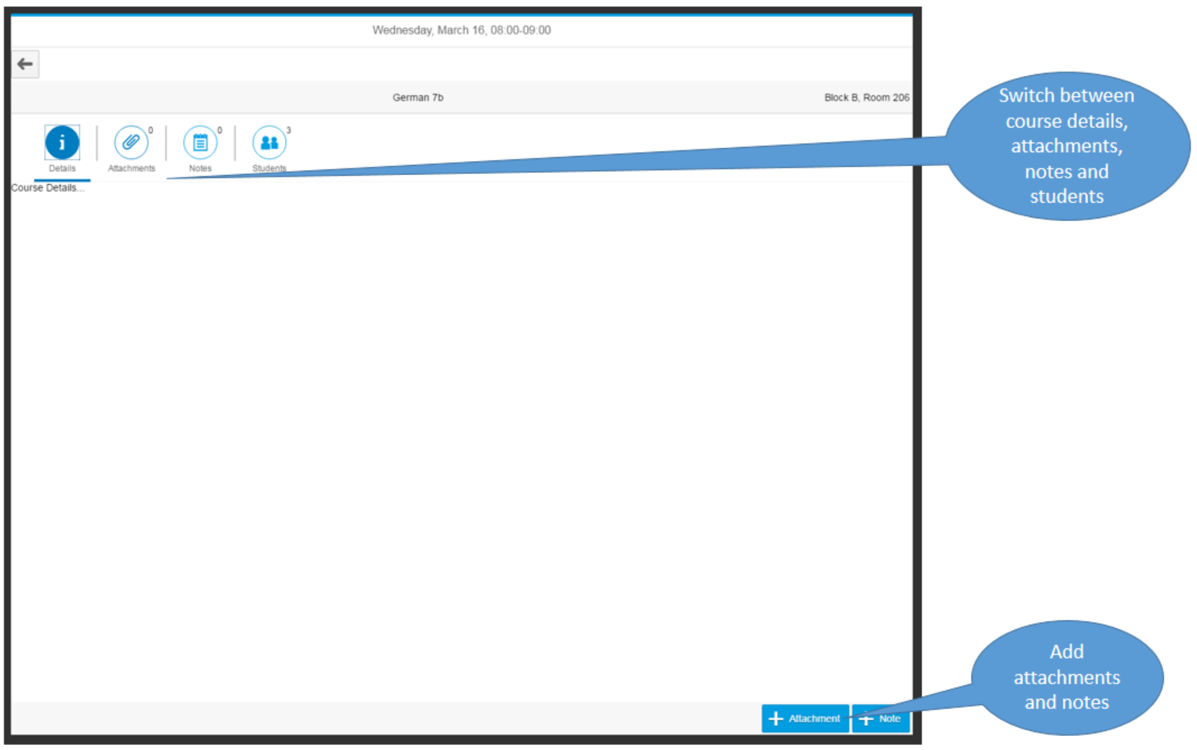
März 2016

Mi	Do	Fr	Sa	So	Mo	Di
16	17	18	19	20	21	22

Schedule for Wednesday, March 16

Time	Course	Room
08:00-09:00	German 7b	**Block B, Room 206**
09:15-10:15	German 8a	**Block A, Room 102**
11:00-12:00	Physics 6b	**Block A, Room 309**

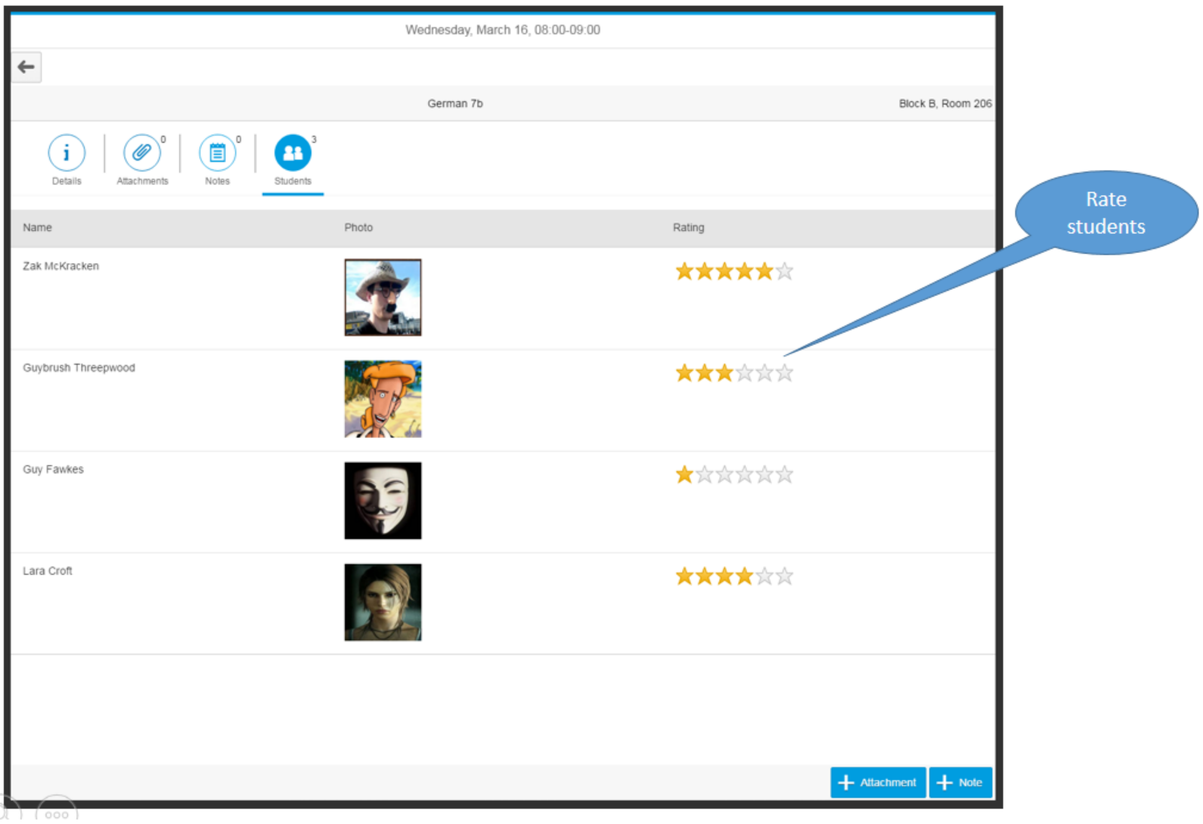
Manage Courses
Select date to view scheduled courses
Click on course to view details
Add, edit, delete courses


Wednesday, March 16, 08:00-09:00

German 7b | Block B, Room 206

Course Details...

Details
Attachments 0
Notes 0
Students 3
Attachment
Note
Switch between course details, attachments, notes and students
Add attachments and notes



Wednesday, March 16, 08:00-09:00

German 7b

Block B, Room 206

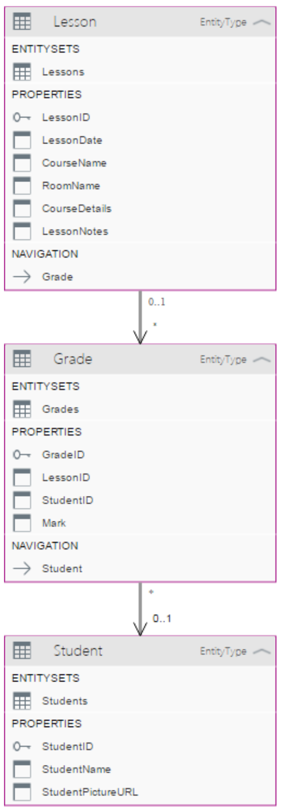
Name	Photo	Rating
Zak McKracken		★★★★★☆
Guybrush Threepwood		★★★☆☆☆
Guy Fawkes		★☆☆☆☆☆
Lara Croft		★★★★☆☆

Study

https://standard.experiencesplash.com/home/projects/c95c424ee55ada6b0baae49a/research/participant/af020a3db87a0f4a0bac3646

Prototype

For my Prototype I started with building a simple data model. I used the OData Model Editor in SAP Web IDE to build an edmx file.

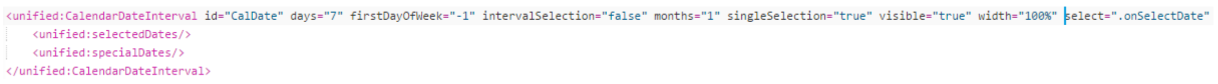


The data model consists of three Entities:

- Lesson: Contains the main data for the lessons. To each lesson, belong 0..* grades to rate the students.
- Grade: Student ratings are saved here. Each grade belongs to one student.
- Student: Students can have 0..* grades.

Of course, for a real Application the data model would have to be much more complex. But for the prototype, this should be sufficient.

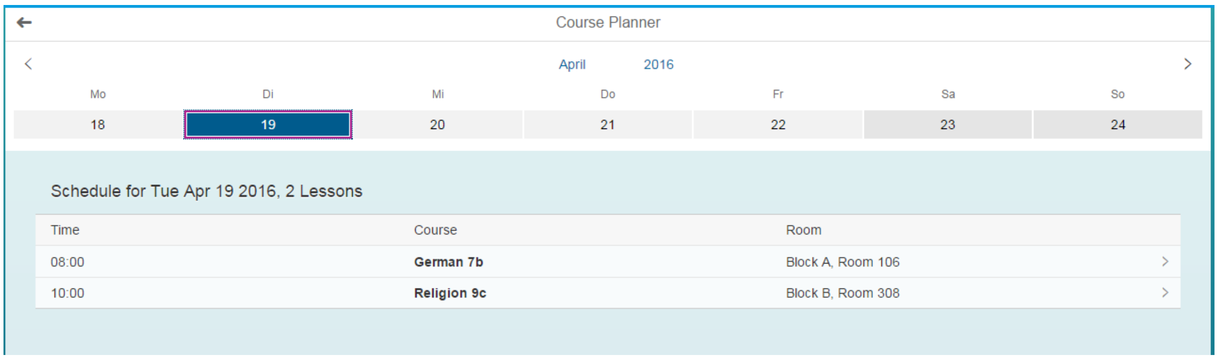
With the data model set up, I needed to find a good starting point for the app. I decided to create the project from the template “Fiori worklist application”, as this seemed to be the best match for mockup. I used the edmx-file to provide the service information for the wizard and selected lessons as the Object Collection for the data binding.
That created a basic worklist view and an object view to start development. Navigation between the two views was already implemented by the wizard.
In my mockup, I used a Calendar Date Interval Control to select dates. This control isn´t supported by the Layout-Editor yet, so I added it to the worklist view in the code editor:



```
<unified:CalendarDateInterval id="CalDate" days="7" firstDayOfWeek="-1" intervalSelection="false" months="1" singleSelection="true" visible="true" width="100%" select=".onSelectDate"
    <unified:selectedDates/>
    <unified:specialDates/>
</unified:CalendarDateInterval>
```

When selecting a date, the method onSelectDate in the worklist controller is called, that filters the worklist-table to show the corresponding lessons:

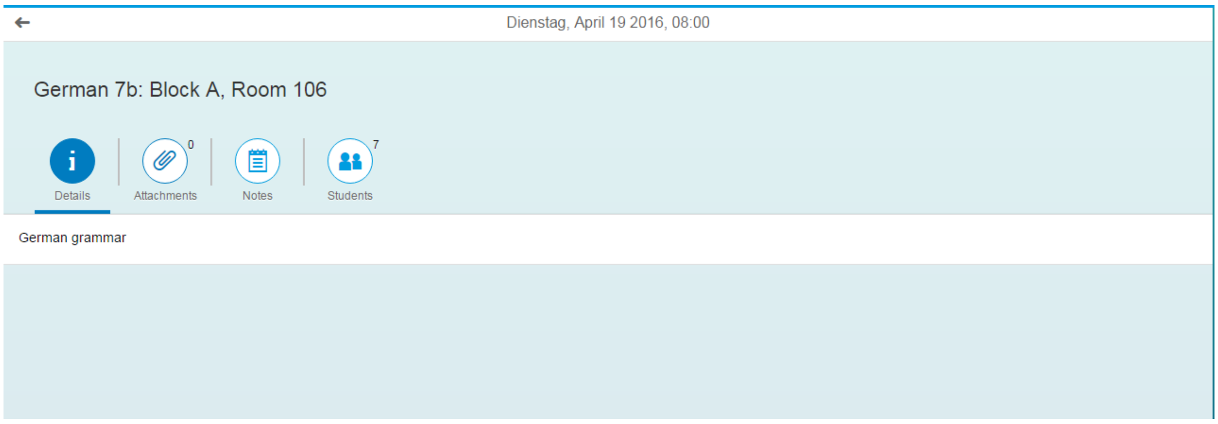
```
onSelectDate : function (oEvent) {
    // The source is the list item that got pressed
    var calendar = oEvent.getSource();
    var dateRange = calendar.getSelectedDates();
    var startDate = dateRange[0].getStartDate();
    var endDate = new Date(startDate.getTime());
    endDate.setHours(23);
    endDate.setMinutes(59);
    endDate.setSeconds(59);

    var aFilter = [];
    aFilter = [new Filter("LessonDate", FilterOperator.BT, startDate, endDate)];
    this._oTable.getBinding("items").filter(aFilter, "Application");
    var sTitle;
    sTitle = "Schedule for " + startDate.toDateString();
    this.getModel("worklistView").setProperty("/worklistTableTitle", sTitle);
},
```



Schedule for Tue Apr 19 2016, 2 Lessons

Time	Course	Room
08:00	**German 7b**	Block A, Room 106
10:00	**Religion 9c**	Block B, Room 308

A click into a row opens the object view of the corresponding lesson. That navigation was taken over from the template.

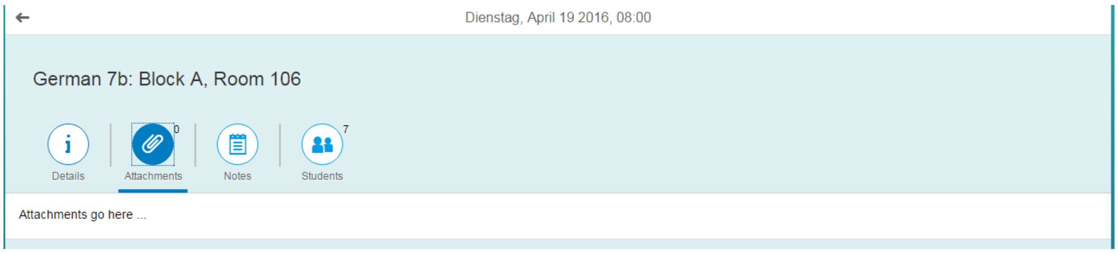


German grammar

In the object view the date of the lesson is displayed in the header. A click on the arrow on the left side of the arrow takes you back to the overview page.
In the object header the course name and the room is displayed.
Below I added an Icon Tab Bar control with the layout editor and included four Icon Tab Filters. With the Icon Tab you can switch between various information and actions for the selected lesson:

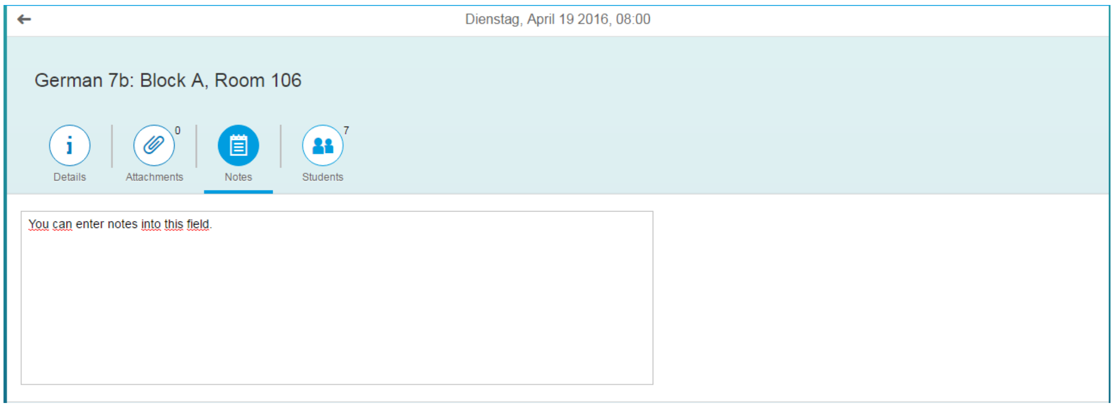
- Details: provides further details for the lesson
- Attachments: this tab is designated to hold various attachments for the lesson. This could be PDF-Files, Videos or else. There is a button in the footer to add attachments. I didn’t

implement this tab in the prototype because of a lack of time.



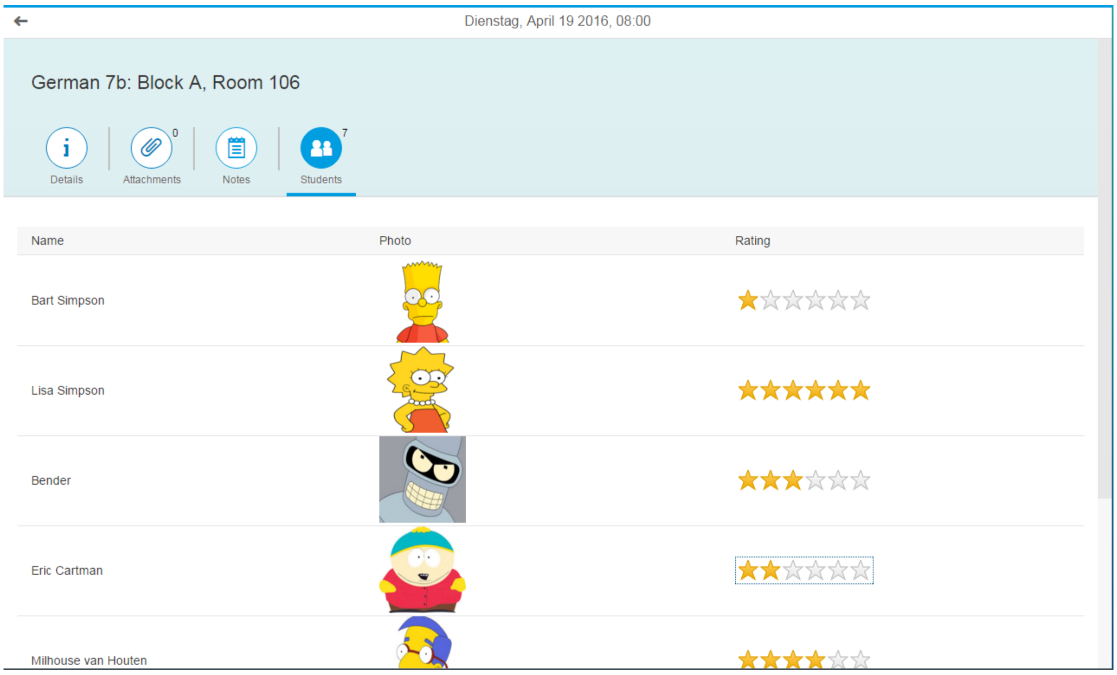
Attachments go here ...

- Notes: You can enter notes in plain text on this tab. I used a Text Area control to display and edit the notes.



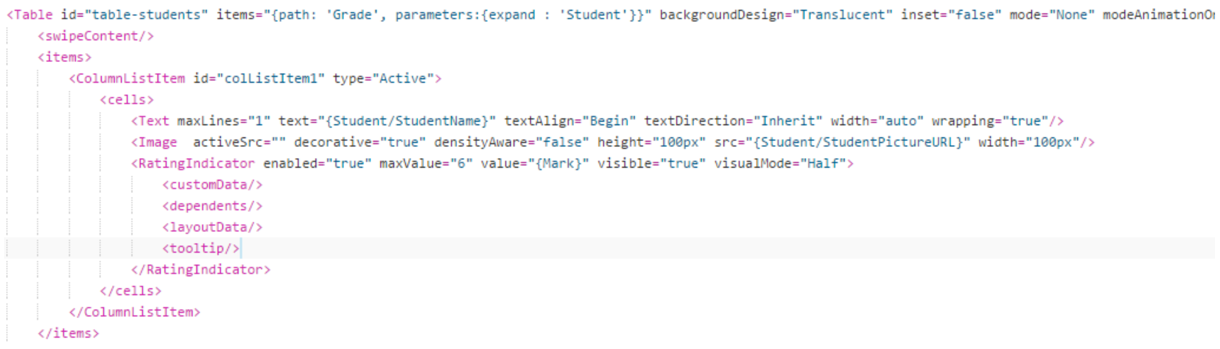
You can enter notes into this field.

- Students: In this tab you get an overview of the students that participate in this lesson. To help identify students, there is a photo for each. The user can also grade the students for the lesson. The number of students is displayed in the Icon Tab Filter.



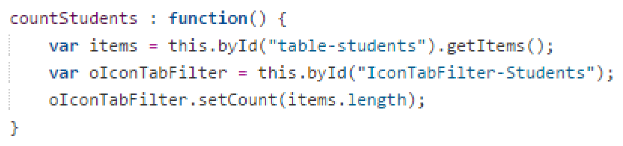
Name	Photo	Rating
Bart Simpson		★☆☆☆☆☆
Lisa Simpson		★★★★★★
Bender		★★★☆☆☆
Eric Cartman		★★☆☆☆☆
Milhouse van Houten		★★★★☆☆

I used a table to display students and grades and bound it to the Grade Navigation property of lessons with an Expand to Student. This fetches the data for the corresponding lesson. This is done by the controller and model generated by the wizard from template and doesn’t have to be implement manually.



```xml
<Table id="table-students" items="{path: 'Grade', parameters:{expand : 'Student'}}" backgroundDesign="Translucent" inset="false" mode="None" modeAnimationO
    <swipeContent/>
    <items>
        <ColumnListItem id="colListItem1" type="Active">
            <cells>
                <Text maxLines="1" text="{Student/StudentName}" textAlign="Begin" textDirection="Inherit" width="auto" wrapping="true"/>
                <Image  activeSrc="" decorative="true" densityAware="false" height="100px" src="{Student/StudentPictureURL}" width="100px"/>
                <RatingIndicator enabled="true" maxValue="6" value="{Mark}" visible="true" visualMode="Half">
                    <customData/>
                    <dependents/>
                    <layoutData/>
                    <tooltip/>
                </RatingIndicator>
            </cells>
        </ColumnListItem>
    </items>
```

To set the counter for the students in the tab filter, I wrote a simple method that is called on the updateFinished-Event of the table.



```javascript
countStudents : function() {
    var items = this.byId("table-students").getItems();
    var oIconTabFilter = this.byId("IconTabFilter-Students");
    oIconTabFilter.setCount(items.length);
}
```

I created mock data for the prototype in the Web IDE. In the real app, there would be a function to enter and edit lesson data. There is a button for this in the footer of the prototype but this isn´t implement yet.

Video of the App Prototype

https://youtu.be/KU4VlvZFmWI