Health
Visitor

CS1 Task 14: Abschluss

FLORIAN AUDERSET, TOBIAS FLÜHMANN,
REMO HÄUSLER, MATHIAS RUDOLF, RAPHAEL SUTER

Abschlusspräsentation

- Vorführung von: Health Visitor
- Reviews der Entwickler
- Gesammelte Erfahrungen

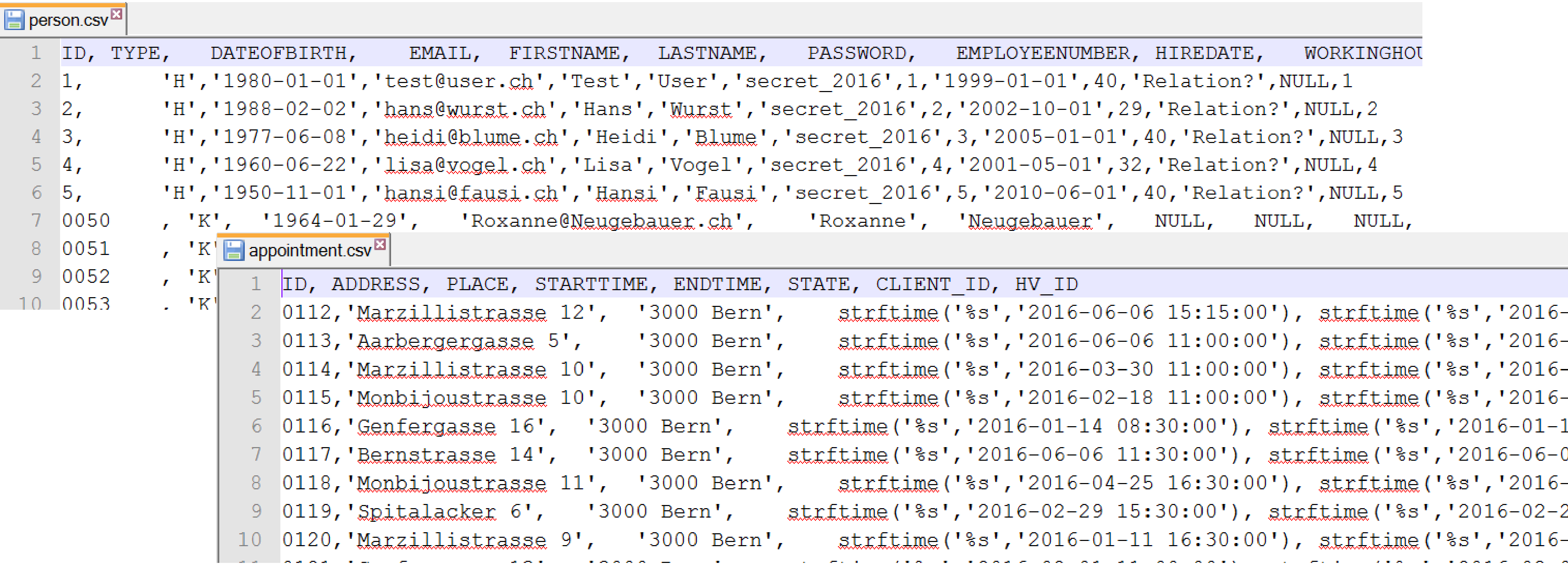
Demo: Health Visitor

- Kurze Vorführung des Prototyps

Persönliches Review der Entwickler

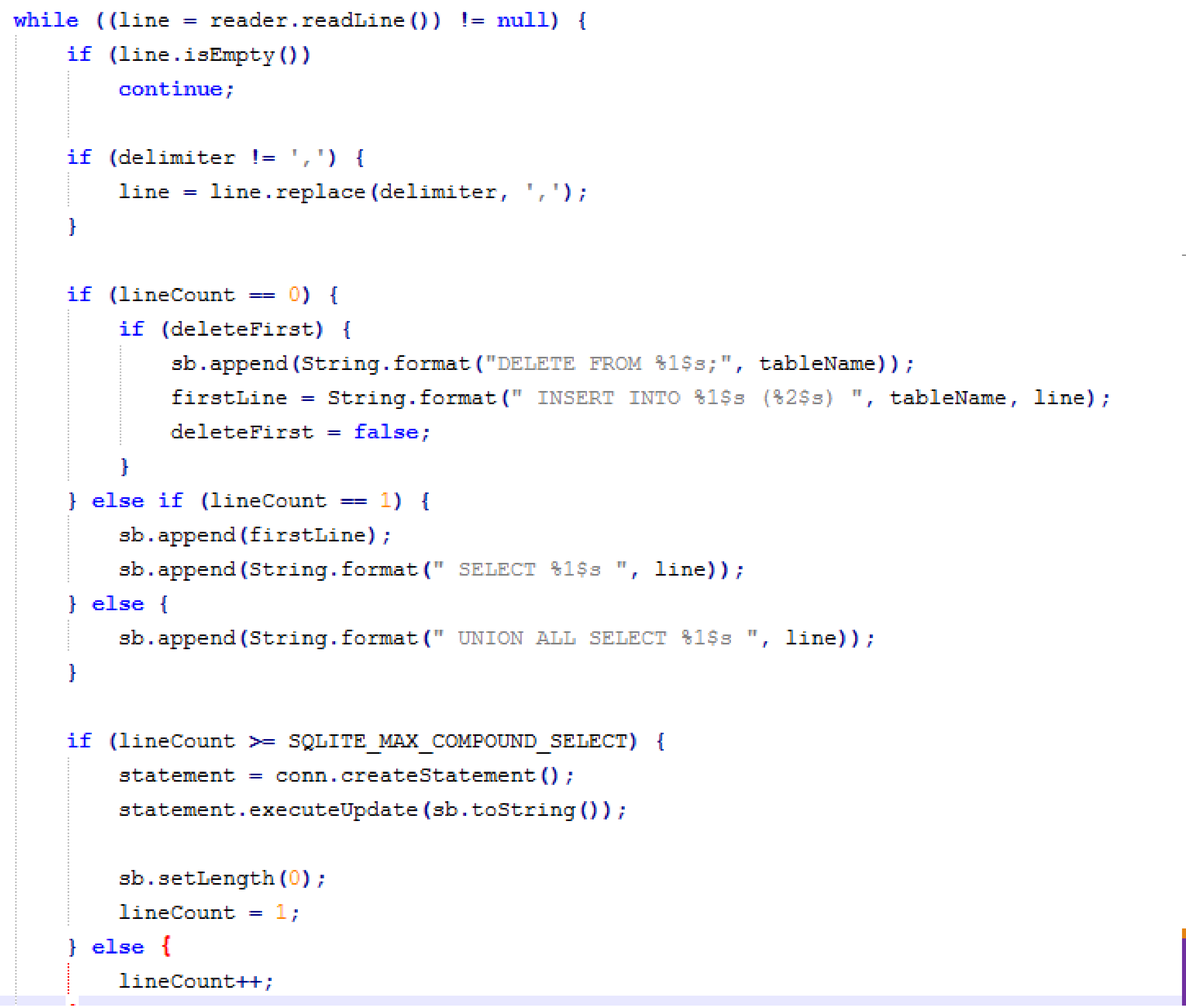
- Mathias Rudolf: Datenaufbereitung & Datenbank
- Tobias Flühmann: Datenspeicherung
- Remo Häusler: Medikation
- Raphael Suter: Rapportierung
- Florian Auderset: UI Design

Datenaufbereitung & Datenbank



```
ID, TYPE,   DATEOFBIRTH,    EMAIL,  FIRSTNAME,  LASTNAME,   PASSWORD,   EMPLOYEENUMBER, HIREDATE,   WORKINGHOU
1,       'H','1980-01-01','test@user.ch','Test','User','secret_2016',1,'1999-01-01',40,'Relation?',NULL,1
2,       'H','1988-02-02','hans@wurst.ch','Hans','Wurst','secret_2016',2,'2002-10-01',29,'Relation?',NULL,2
3,       'H','1977-06-08','heidi@blume.ch','Heidi','Blume','secret_2016',3,'2005-01-01',40,'Relation?',NULL,3
4,       'H','1960-06-22','lisa@vogel.ch','Lisa','Vogel','secret_2016',4,'2001-05-01',32,'Relation?',NULL,4
5,       'H','1950-11-01','hansi@fausi.ch','Hansi','Fausi','secret_2016',5,'2010-06-01',40,'Relation?',NULL,5
0050     , 'K',  '1964-01-29',    'Roxanne@Neugebauer.ch',     'Roxanne',  'Neugebauer',    NULL,    NULL,    NULL,
0051     , 'K'
0052     , 'K'
0053     , 'K'
```

```
ID, ADDRESS, PLACE, STARTTIME, ENDTIME, STATE, CLIENT_ID, HV_ID
0112,'Marzillistrasse 12',   '3000 Bern',    strftime('%s','2016-06-06 15:15:00'), strftime('%s','2016-
0113,'Aarbergergasse 5',     '3000 Bern',    strftime('%s','2016-06-06 11:00:00'), strftime('%s','2016-
0114,'Marzillistrasse 10',   '3000 Bern',    strftime('%s','2016-03-30 11:00:00'), strftime('%s','2016-
0115,'Monbijoustrasse 10',   '3000 Bern',    strftime('%s','2016-02-18 11:00:00'), strftime('%s','2016-
0116,'Genfergasse 16',   '3000 Bern',    strftime('%s','2016-01-14 08:30:00'), strftime('%s','2016-01-1
0117,'Bernstrasse 14',   '3000 Bern',    strftime('%s','2016-06-06 11:30:00'), strftime('%s','2016-06-0
0118,'Monbijoustrasse 11',   '3000 Bern',    strftime('%s','2016-04-25 16:30:00'), strftime('%s','2016-
0119,'Spitalacker 6',    '3000 Bern',    strftime('%s','2016-02-29 15:30:00'), strftime('%s','2016-02-2
0120,'Marzillistrasse 9',    '3000 Bern',    strftime('%s','2016-01-11 16:30:00'), strftime('%s','2016-
```



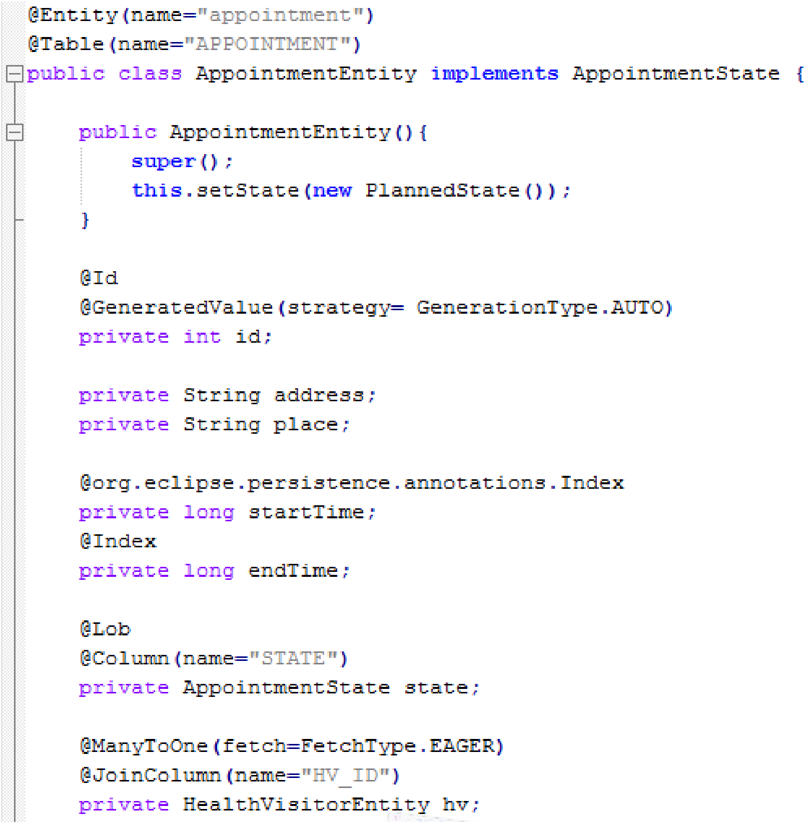
```
while ((line = reader.readLine()) != null) {
    if (line.isEmpty())
        continue;

    if (delimiter != ',') {
        line = line.replace(delimiter, ',');
    }

    if (lineCount == 0) {
        if (deleteFirst) {
            sb.append(String.format("DELETE FROM %1$s;", tableName));
            firstLine = String.format(" INSERT INTO %1$s (%2$s) ", tableName, line);
            deleteFirst = false;
        }
    } else if (lineCount == 1) {
        sb.append(firstLine);
        sb.append(String.format(" SELECT %1$s ", line));
    } else {
        sb.append(String.format(" UNION ALL SELECT %1$s ", line));
    }

    if (lineCount >= SQLITE_MAX_COMPOUND_SELECT) {
        statement = conn.createStatement();
        statement.executeUpdate(sb.toString());

        sb.setLength(0);
        lineCount = 1;
    } else {
        lineCount++;
```

Datenspeicherung

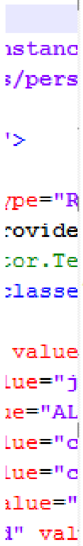
- Datenbank oder Files
- Datenbank → SQL
- Datenbankevaluation
  - Online Instanz MySQL
  - Lokale sqlite3 DB

- JPA
- Evaluation
  - Eclipselink
  - Hibernate
  - Apache Open JPA

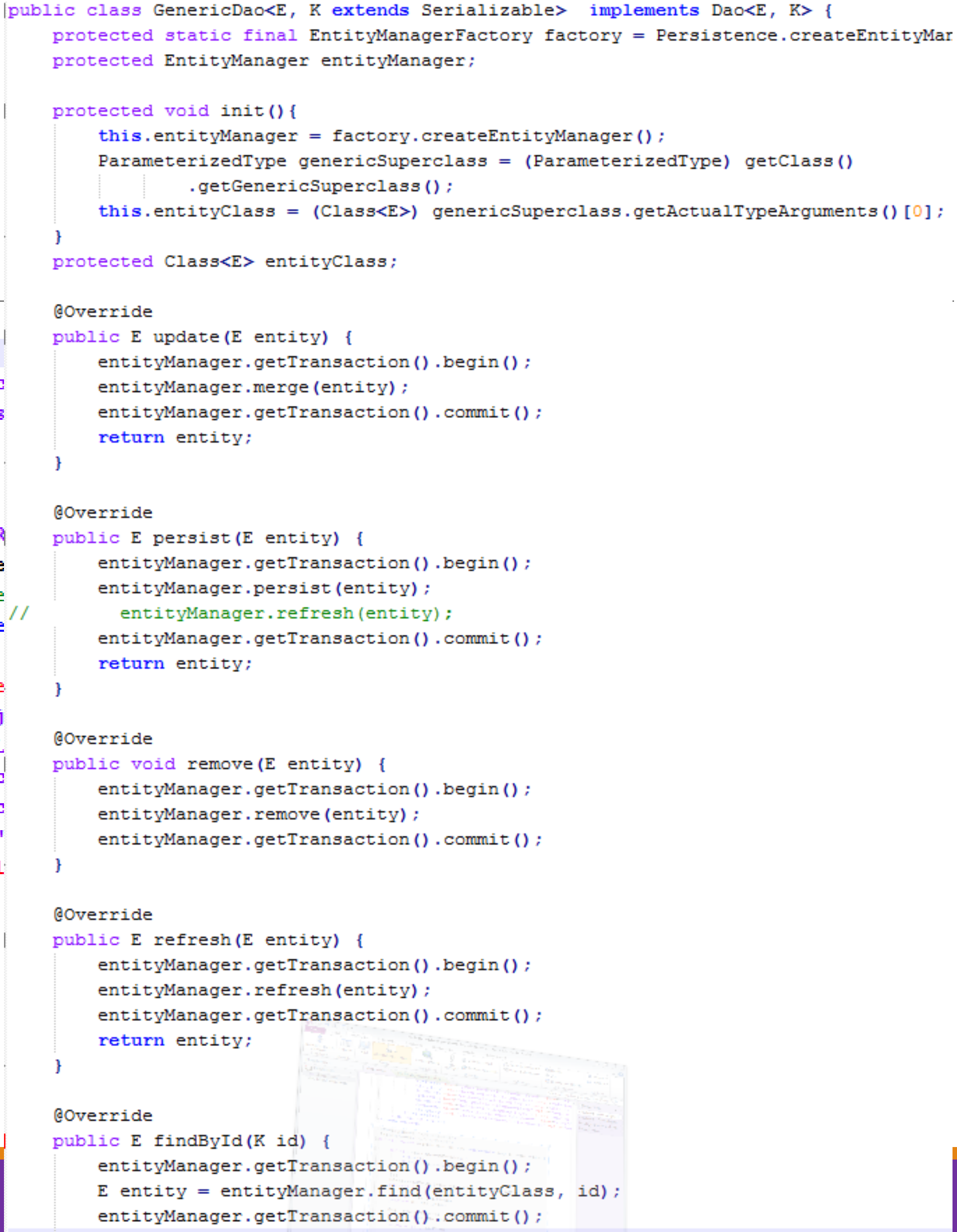
Datenspeicherung



```
@Entity(name="appointment")
@Table(name="APPOINTMENT")
public class AppointmentEntity implements AppointmentState {

    public AppointmentEntity(){
        super();
        this.setState(new PlannedState());
    }

    @Id
    @GeneratedValue(strategy= GenerationType.AUTO)
    private int id;

    private String address;
    private String place;

    @org.eclipse.persistence.annotations.Index
    private long startTime;
    @Index
    private long endTime;

    @Lob
    @Column(name="STATE")
    private AppointmentState state;

    @ManyToOne(fetch=FetchType.EAGER)
    @JoinColumn(name="HV_ID")
    private HealthVisitorEntity hv;
```

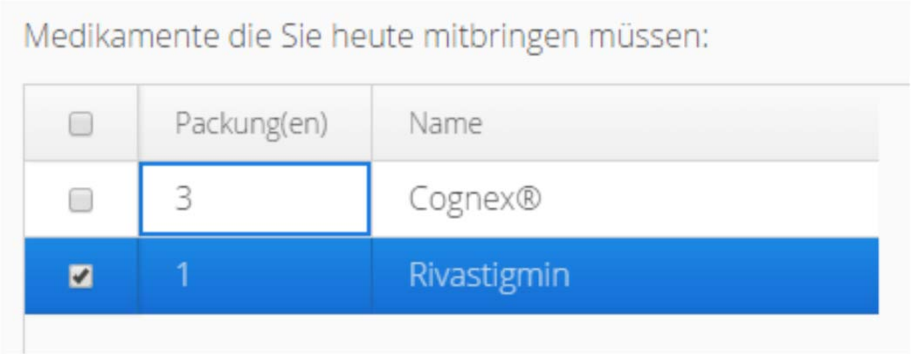


```
ıstanc
;/pers

'>

/pe="R
rovide
:or.Te
:lasse

 value
lue="j
ıe="AL
lue="c
lue="c
alue="
ı" val
```



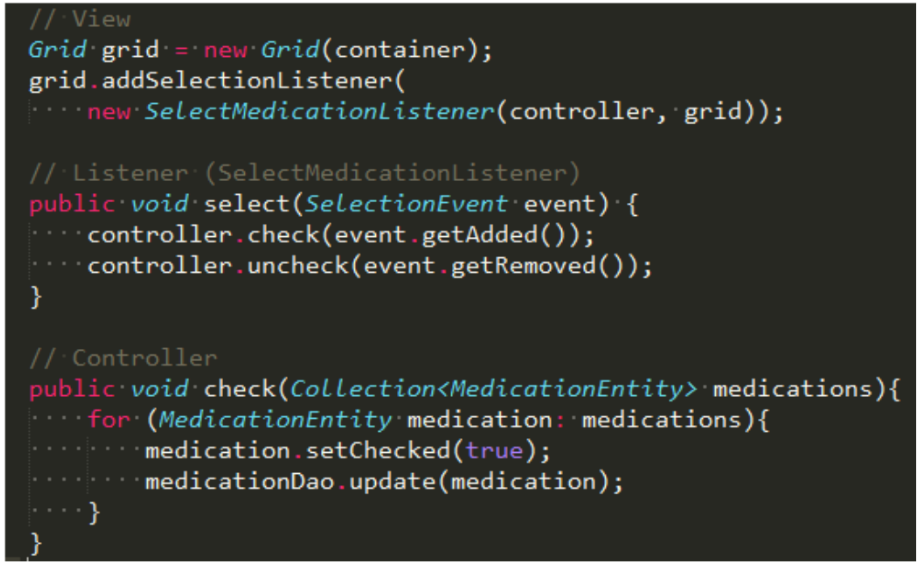
```
public class GenericDao<E, K extends Serializable>  implements Dao<E, K> {
    protected static final EntityManagerFactory factory = Persistence.createEntityMar
    protected EntityManager entityManager;

    protected void init(){
        this.entityManager = factory.createEntityManager();
        ParameterizedType genericSuperclass = (ParameterizedType) getClass()
                .getGenericSuperclass();
        this.entityClass = (Class<E>) genericSuperclass.getActualTypeArguments()[0];
    }
    protected Class<E> entityClass;

    @Override
    public E update(E entity) {
        entityManager.getTransaction().begin();
        entityManager.merge(entity);
        entityManager.getTransaction().commit();
        return entity;
    }

    @Override
    public E persist(E entity) {
        entityManager.getTransaction().begin();
        entityManager.persist(entity);
//        entityManager.refresh(entity);
        entityManager.getTransaction().commit();
        return entity;
    }

    @Override
    public void remove(E entity) {
        entityManager.getTransaction().begin();
        entityManager.remove(entity);
        entityManager.getTransaction().commit();
    }

    @Override
    public E refresh(E entity) {
        entityManager.getTransaction().begin();
        entityManager.refresh(entity);
        entityManager.getTransaction().commit();
        return entity;
    }

    @Override
    public E findById(K id) {
        entityManager.getTransaction().begin();
        E entity = entityManager.find(entityClass, id);
        entityManager.getTransaction().commit();
```

Medikation



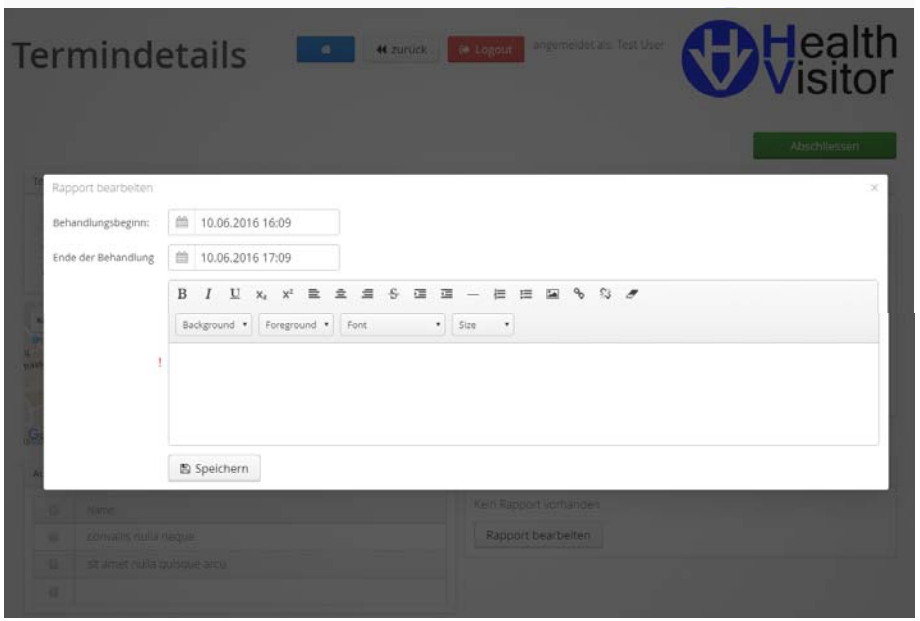
Medikamente die Sie heute mitbringen müssen:

□	Packung(en)	Name
□	3	Cognex®
☑	1	Rivastigmin



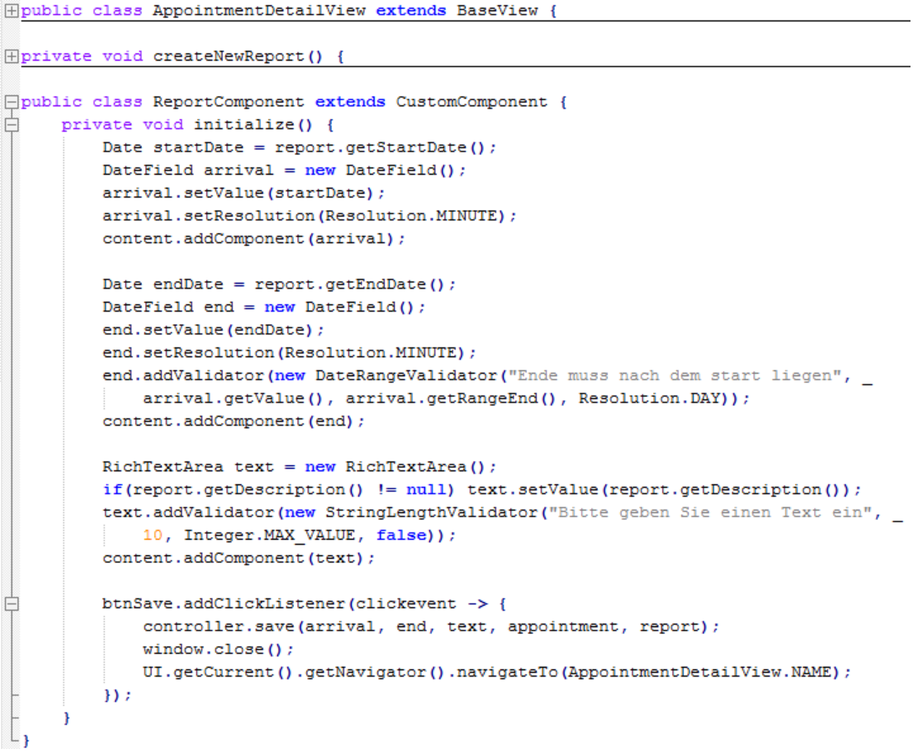
```
// View
Grid grid = new Grid(container);
grid.addSelectionListener(
    new SelectMedicationListener(controller, grid));

// Listener (SelectMedicationListener)
public void select(SelectionEvent event) {
    controller.check(event.getAdded());
    controller.uncheck(event.getRemoved());
}

// Controller
public void check(Collection<MedicationEntity> medications){
    for (MedicationEntity medication: medications){
        medication.setChecked(true);
        medicationDao.update(medication);
    }
}
```

Rapportierung



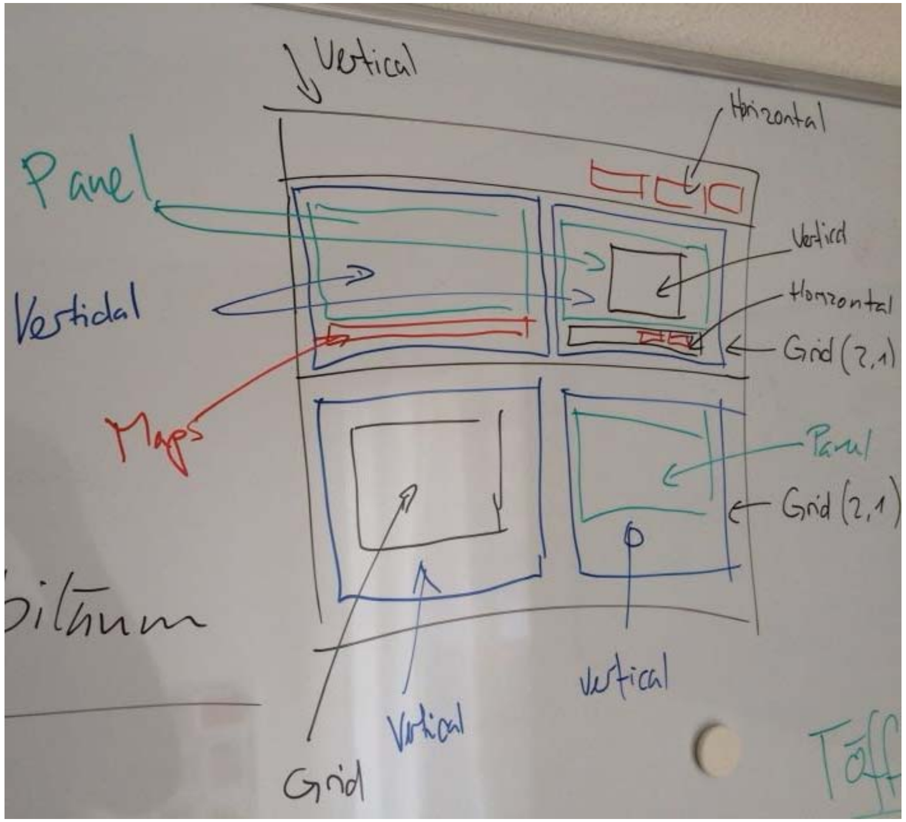
Behandlungsbeginn: 10.06.2016 16:09

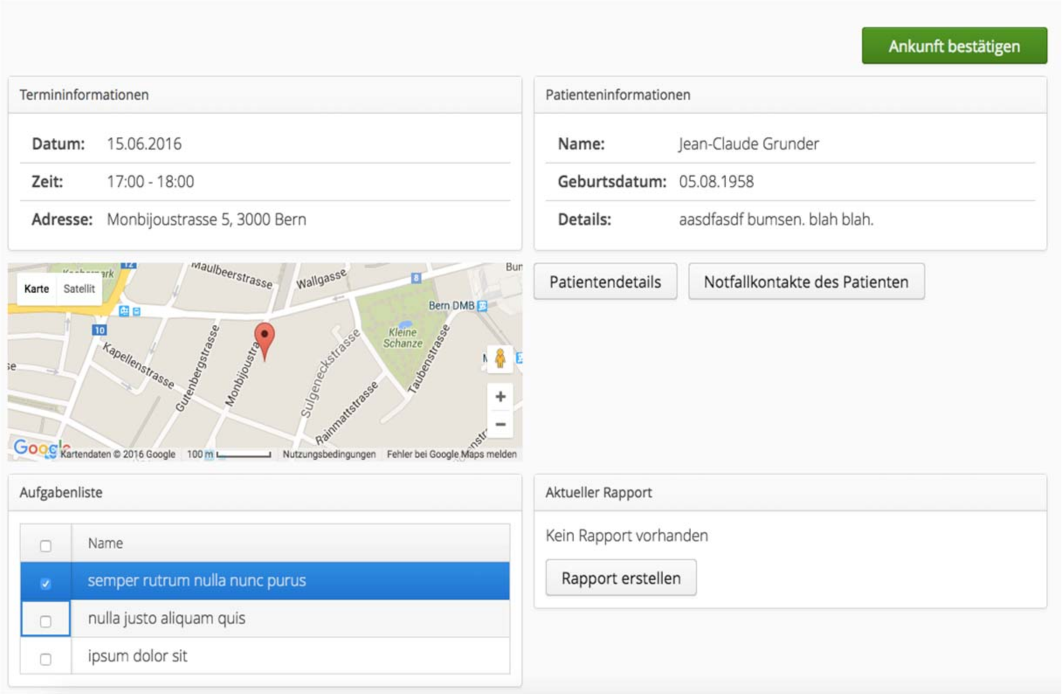
Ende der Behandlung 10.06.2016 17:09



```
public class AppointmentDetailView extends BaseView {

private void createNewReport() {

public class ReportComponent extends CustomComponent {
    private void initialize() {
        Date startDate = report.getStartDate();
        DateField arrival = new DateField();
        arrival.setValue(startDate);
        arrival.setResolution(Resolution.MINUTE);
        content.addComponent(arrival);

        Date endDate = report.getEndDate();
        DateField end = new DateField();
        end.setValue(endDate);
        end.setResolution(Resolution.MINUTE);
        end.addValidator(new DateRangeValidator("Ende muss nach dem start liegen", _
            arrival.getValue(), arrival.getRangeEnd(), Resolution.DAY));
        content.addComponent(end);

        RichTextArea text = new RichTextArea();
        if(report.getDescription() != null) text.setValue(report.getDescription());
        text.addValidator(new StringLengthValidator("Bitte geben Sie einen Text ein", _
            10, Integer.MAX_VALUE, false));
        content.addComponent(text);

        btnSave.addClickListener(clickevent -> {
            controller.save(arrival, end, text, appointment, report);
            window.close();
            UI.getCurrent().getNavigator().navigateTo(AppointmentDetailView.NAME);
        });
    }
}
```

UI Design





Termininformationen

Datum:	15.06.2016
Zeit:	17:00 - 18:00
Adresse:	Monbijoustrasse 5, 3000 Bern

Patienteninformationen

Name:	Jean-Claude Grunder
Geburtsdatum:	05.08.1958
Details:	aasdfasdf bumsen. blah blah.

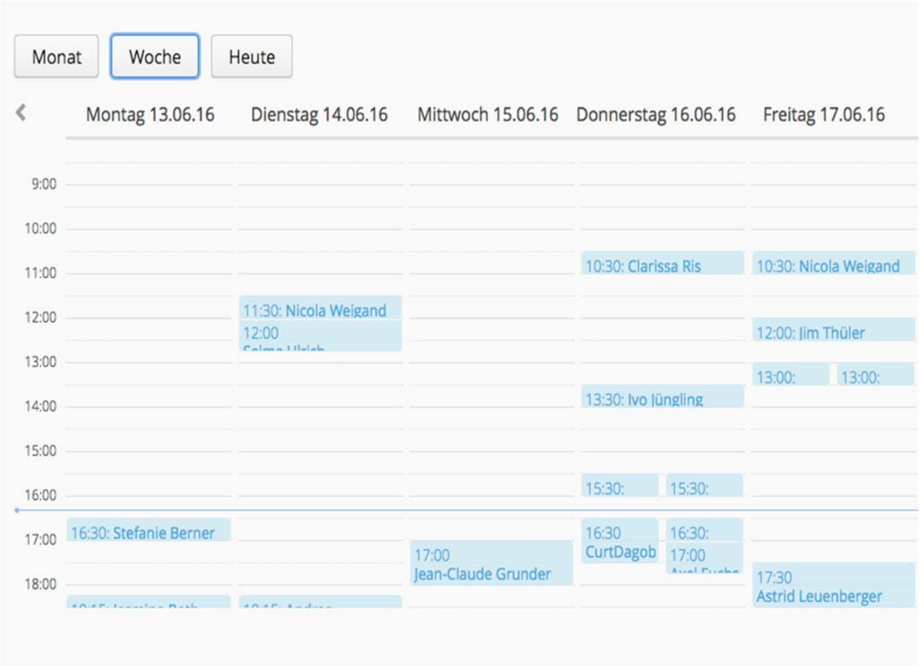
Aufgabenliste

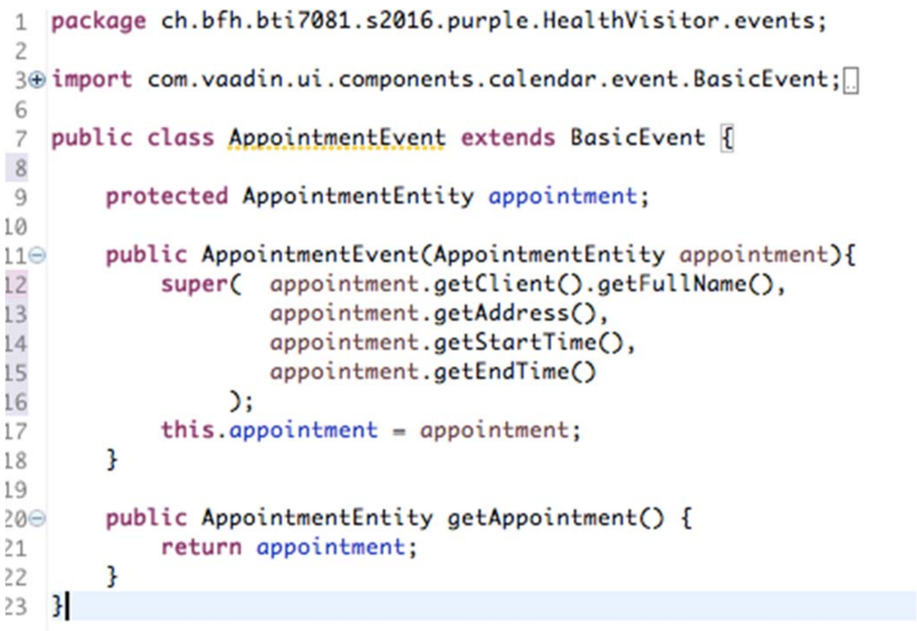
□	Name
☑	semper rutrum nulla nunc purus
□	nulla justo aliquam quis
□	ipsum dolor sit

Aktueller Rapport

Kein Rapport vorhanden

UI Design



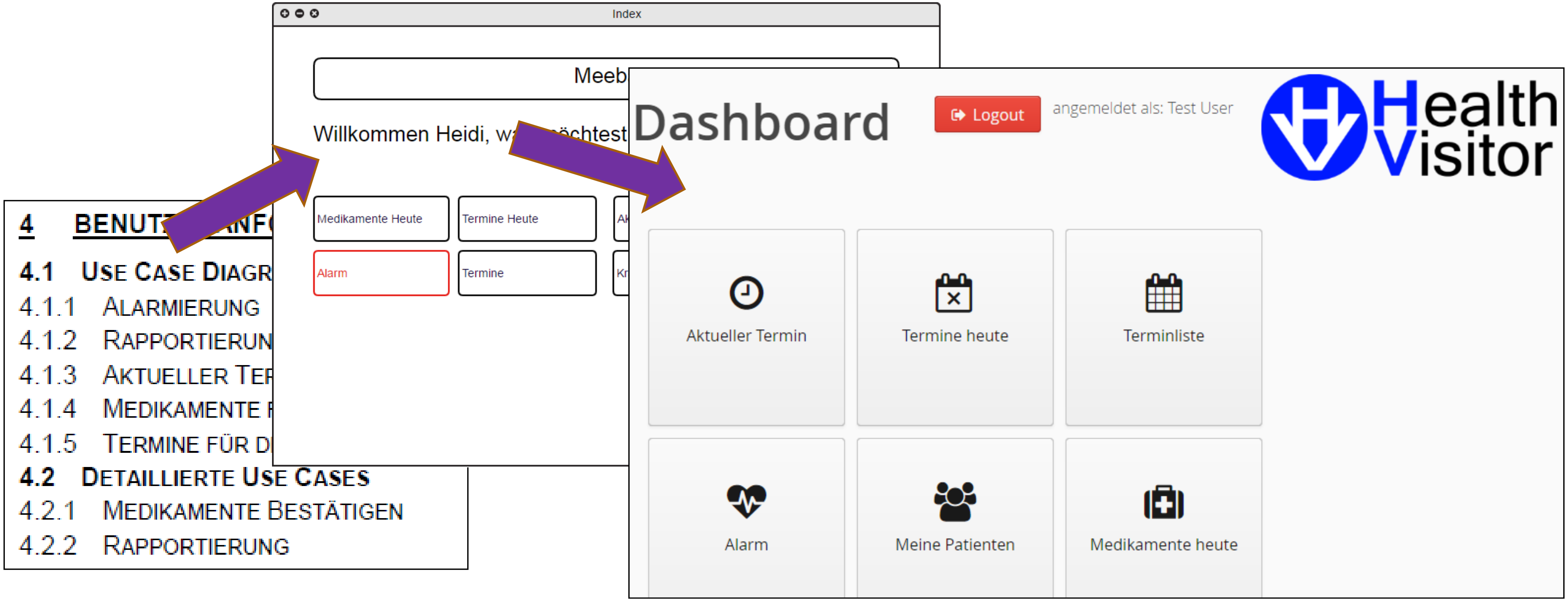


```java
package ch.bfh.bti7081.s2016.purple.HealthVisitor.events;

import com.vaadin.ui.components.calendar.event.BasicEvent;

public class AppointmentEvent extends BasicEvent {

    protected AppointmentEntity appointment;

    public AppointmentEvent(AppointmentEntity appointment){
        super(  appointment.getClient().getFullName(),
                appointment.getAddress(),
                appointment.getStartTime(),
                appointment.getEndTime()
              );
        this.appointment = appointment;
    }

    public AppointmentEntity getAppointment() {
        return appointment;
    }
}
```

Gesammelte Erfahrungen

- Von den Storyboards zur Implementation
- Rückblick auf die Entwicklung mit Scrum
- Zusammenarbeit im Team
- Allgemeiner Rückblick

Von den Storyboards zur Implementation



Willkommen Heidi, w öchtest

Dashboard

Entwicklung mit Scrum

- Scrum Master
- Backlogs
- Planung der Sprints inkl. Zeitmanagement

Zusammenarbeit im Team

- Einer für alle, alle für einen.
- Probleme wurden im Daily Scrum angesprochen und konnten sofort an andere weitergegeben werden.
- Agile Verantwortung der Tasks
- Skype!

Allgemeiner Rückblick