Entwicklung einer Formularanwendung mit Kompatibilitätsvalidierung der Einfach- und Mehrfachauswahl-Eingabefelder

Vorgelegt von:

Alexander Johr

Meine Adresse

Erstprüfer:	Prof. Jürgen Singer Ph.D.
Zweitprüfer:	Prof. Daniel Ackermann
Datum:	02.11.2020

Teil I

Implementierung

0.1 Schritt 4

```dart
class SelectionCard<ChoiceType extends Choice> extends StatelessWidget {
  final String title;
  final BehaviorSubject<BuiltSet<ChoiceType>> selectionViewModel;
  final Choices<ChoiceType> allChoices;
  final BehaviorSubject<Set<Choice>> priorChoices;
  final OnSelect<ChoiceType> onSelect;
  final OnDeselect<ChoiceType> onDeselect;
  final String? errorText;

  SelectionCard(
      {required this.title,
      required Iterable<ChoiceType> initialValue,
      required this.allChoices,
      required this.priorChoices,
      required this.onSelect,
      required this.onDeselect,
      this.errorText,
      Key? key})
```

Listing 1: XXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/widgets/selection_card.dart

```dart
return StreamBuilder(
    stream: selectionViewModel,
    builder: (context, snapshot) {
      final selectedChoices = selectionViewModel.value;
      final bool wrongSelection = selectedChoices
          .any((c) => !c.conditionMatches(priorChoices.value));

      return Card(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            ListTile(
              focusNode: focusNode,
              title: Text(title),
              subtitle: Text(
                  selectedChoices.map((c) => c.description).join(", ")),
              trailing: const Icon(Icons.edit),
              onTap: navigateToSelectionScreen,
              tileColor:
                  wrongSelection || errorText != null ? Colors.red : null,
```

Listing 2: XXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/widgets/selection_card.dart

```
Set<ChoiceType> selectedAndSelectableChoices = {};
Set<ChoiceType> unselectableChoices = {};

for (ChoiceType c in allChoices) {
  if (selectedChoices.contains(c) ||
      c.conditionMatches(priorChoices.value)) {
    selectedAndSelectableChoices.add(c);
  } else {
    unselectableChoices.add(c);
  }
}

return ListView(children: [
  ...selectedAndSelectableChoices.map((ChoiceType c) {
    bool isSelected = selectedChoices.contains(c);
    bool selectedButDoesNotMatch =
        isSelected && !c.conditionMatches(priorChoices.value);
```

Listing 3: XXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/widgets/selection_card.dart

```
return CheckboxListTile(
    key: Key(
        "valid choice ${allChoices.name} - ${c.abbreviation}"),
    controlAffinity: ListTileControlAffinity.leading,
    title: Text(c.description),
```

Listing 4: XXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/widgets/selection_card.dart

```
onPressed: () => Navigator.of(context).pop(),
```

Listing 5: XXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/widgets/selection_card.dart

```
      ),
    );
  }
}
```

Listing 6: XXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/widgets/selection_card.dart

```
BehaviorSubject<Set<Choice>> priorChoices =
    BehaviorSubject<Set<Choice>>.seeded({});

MassnahmenFormViewModel() {
  Stream<Set<Choice>> choicesStream = Rx.combineLatest([
    foerderklasse,
    kategorie,
    zielflaeche,
    zieleinheit,
    hauptzielsetzungLand,
  ], (_) {
    return {
      if (foerderklasse.value != null) foerderklasse.value!,
      if (kategorie.value != null) kategorie.value!,
      if (zielflaeche.value != null) zielflaeche.value!,
      if (zieleinheit.value != null) zieleinheit.value!,
      if (hauptzielsetzungLand.value != null) hauptzielsetzungLand.value!,
    };
  });

  choicesStream.listen((event) => priorChoices.add(event));
}
```

Listing 7: XXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/screens/massnahmen_detail/massnahmen_form_view_model.dart

```
    choices.any((c) => !c.conditionMatches(vm.priorChoices.value));

if (atLeastOneValueInvalid) {
  return "Wenigstens ein Wert im Feld ${allChoices.name} enthält ist fehlerhaft!";
```

Listing 8: Die Ausgabe der Formularfelder, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/screens/massnahmen_detail/massnahmen_detail.dart

```
            },
            onSelect: (selectedChoice) =>
                selectionViewModel.value = selectedChoice,
            onDeselect: (selectedChoice) => selectionViewModel.value = null,
            errorText: field.errorText,
          ));
}

return Scaffold(
  appBar: AppBar(
```

Listing 9: Die Ausgabe der Formularfelder, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/screens/massnahmen_detail/massnahmen_detail.dart

```
onWillPop: () {
  if (inputsAreValidOrNotMarkedFinal()) {
    saveRecord();
    return Future.value(true);
```

Listing 10: Die Ausgabe der Formularfelder, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/screens/massnahmen_detail/massnahmen_detail.dart

```dart
typedef Condition = bool Function(Set<Choice> choices);

class Choice {
  final String description;
  final String abbreviation;
  final bool Function(Set<Choice> choices) condition;

  bool conditionMatches(Set<Choice> choices) => condition.call(choices);

  bool conditionDoesNotMatch(Set<Choice> choices) => !condition.call(choices);

  const Choice(this.abbreviation, this.description, {Condition? condition})
      : condition = condition ?? _conditionIsAlwaysMet;

  static bool _conditionIsAlwaysMet(Set<Choice> choices) => true;
}
```

Listing 11: XXXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/choices/base/choice.dart

```dart
static final al = ZielflaecheChoice("al", "AL",
    condition: (choices) => !choices.contains(KategorieChoice.zf_us));
```

Listing 12: XXXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/choices/choices.dart

```dart
static final wald = ZielflaecheChoice("wald", "Wald/Forst",
    condition: (choices) =>
        (choices.contains(FoerderklasseChoice.ea) ||
            choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
            choices.contains(FoerderklasseChoice.aukm_ohne_vns)) &&
        (!choices.contains(KategorieChoice.zf_us) ||
            !choices.contains(KategorieChoice.bes_kult_rass)));
```

Listing 13: XXXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/choices/choices.dart

```dart
class KategorieChoice extends Choice {
  static final zf_us = KategorieChoice(
      "zf_us", "Anbau Zwischenfrucht/Untersaat",
      condition: (choices) =>
          choices.contains(FoerderklasseChoice.aukm_ohne_vns));
  static final anlage_pflege = KategorieChoice(
      "anlage_pflege", "Anlage/Pflege Struktur",
      condition: (choices) =>
          choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
          choices.contains(FoerderklasseChoice.aukm_ohne_vns));
  static final dungmang = KategorieChoice("dungmang", "Düngemanagement",
      condition: (choices) =>
          choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
          choices.contains(FoerderklasseChoice.aukm_ohne_vns));
  static final extens = KategorieChoice("extens", "Extensivierung");
  static final flst = KategorieChoice("flst", "Flächenstilllegung/Brache",
      condition: (choices) =>
          choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
          choices.contains(FoerderklasseChoice.aukm_ohne_vns));
  static final umwandlg = KategorieChoice("umwandlg", "Nutzungsumwandlung",
      condition: (choices) =>
          choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
          choices.contains(FoerderklasseChoice.aukm_ohne_vns));
  static final bes_kult_rass = KategorieChoice(
      "bes_kult_rass", "Förderung bestimmter Rassen / Sorten / Kulturen",
      condition: (choices) => !choices.contains(FoerderklasseChoice.ea));
  static final contact = KategorieChoice("contact", "bitte um Unterstützung");

  KategorieChoice(String abbreviation, String description,
      {bool Function(Set<Choice> choices)? condition})
      : super(abbreviation, description, condition: condition);
}
```

Listing 14: XXXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/choices/choices.dart

```dart
class ZielflaecheChoice extends Choice {
  static final ka = ZielflaecheChoice("ka", "keine Angabe/Vorgabe");
  static final al = ZielflaecheChoice("al", "AL",
      condition: (choices) => !choices.contains(KategorieChoice.zf_us));
  static final gl = ZielflaecheChoice("gl", "GL");
  static final lf = ZielflaecheChoice("lf", "LF");
  static final dk_sk = ZielflaecheChoice("dk_sk", "DK/SK",
      condition: (choices) => !choices.contains(FoerderklasseChoice.twm_ziel));
  static final hff = ZielflaecheChoice("hff", "HFF");
  static final biotop_le = ZielflaecheChoice(
      "biotop_le", "Landschaftselement/Biotop o.Ä.",
      condition: (choices) =>
          (choices.contains(FoerderklasseChoice.azl) ||
              choices.contains(FoerderklasseChoice.ea) ||
              choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
              choices.contains(FoerderklasseChoice.aukm_ohne_vns)) &&
          (!choices.contains(KategorieChoice.zf_us) ||
              !choices.contains(KategorieChoice.bes_kult_rass)));
  static final wald = ZielflaecheChoice("wald", "Wald/Forst",
      condition: (choices) =>
          (choices.contains(FoerderklasseChoice.ea) ||
              choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
              choices.contains(FoerderklasseChoice.aukm_ohne_vns)) &&
          (!choices.contains(KategorieChoice.zf_us) ||
              !choices.contains(KategorieChoice.bes_kult_rass)));
  static final contact = ZielflaecheChoice("contact", "bitte um Unterstützung");

  ZielflaecheChoice(String abbreviation, String description,
      {bool Function(Set<Choice> choices)? condition})
      : super(abbreviation, description, condition: condition);
}
```

Listing 15: XXXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/choices/choices.dart

```
121 class ZieleinheitChoice extends Choice {
122   static final ka = ZieleinheitChoice("ka", "keine Angabe/Vorgabe");
123   static final m3 = ZieleinheitChoice("m3", "m³ (z.B. Gülle)",
124       condition: (choices) =>
125           (choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
126               choices.contains(FoerderklasseChoice.aukm_ohne_vns)) &&
127           (choices.contains(KategorieChoice.dungmang) ||
128               choices.contains(KategorieChoice.extens)) &&
129           (!choices.contains(ZielflaecheChoice.ka) &&
130               !choices.contains(ZielflaecheChoice.contact)));
131   static final pieces = ZieleinheitChoice(
132       "pieces", "Kopf/Stück (z.B. Tiere oder Bäume)",
133       condition: (choices) =>
134           (choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
135               choices.contains(FoerderklasseChoice.aukm_ohne_vns) ||
136               choices.contains(FoerderklasseChoice.twm_ziel)) &&
137           (!choices.contains(KategorieChoice.zf_us) ||
138               !choices.contains(KategorieChoice.flst) ||
139               !choices.contains(KategorieChoice.umwandlg)) &&
140           (!choices.contains(ZielflaecheChoice.ka) &&
141               !choices.contains(ZielflaecheChoice.contact)));
142   static final gve = ZieleinheitChoice("gve", "GV/GVE",
143       condition: (choices) =>
144           (choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
145               choices.contains(FoerderklasseChoice.aukm_ohne_vns) ||
146               choices.contains(FoerderklasseChoice.twm_ziel)) &&
147           (!choices.contains(KategorieChoice.zf_us) ||
148               !choices.contains(KategorieChoice.anlage_pflege) ||
149               !choices.contains(KategorieChoice.flst) ||
150               !choices.contains(KategorieChoice.umwandlg)) &&
151           (!choices.contains(ZielflaecheChoice.ka) &&
152               !choices.contains(ZielflaecheChoice.contact)));
153   static final rgve = ZieleinheitChoice("rgve", "RGV",
154       condition: (choices) =>
155           (choices.contains(FoerderklasseChoice.aukm_nur_vns) ||
156               choices.contains(FoerderklasseChoice.aukm_ohne_vns) ||
157               choices.contains(FoerderklasseChoice.twm_ziel)) &&
158           (!choices.contains(KategorieChoice.zf_us) ||
159               !choices.contains(KategorieChoice.anlage_pflege) ||
160               !choices.contains(KategorieChoice.flst) ||
161               !choices.contains(KategorieChoice.umwandlg)) &&
162           (!choices.contains(ZielflaecheChoice.ka) &&
163               !choices.contains(ZielflaecheChoice.contact)));
164   static final ha = ZieleinheitChoice("ha", "ha",
165       condition: (choices) =>
166           !choices.contains(ZielflaecheChoice.ka) &&
167           !choices.contains(ZielflaecheChoice.contact));
168   static final contact = ZieleinheitChoice("contact", "bitte um Unterstützung");
169
170   ZieleinheitChoice(String abbreviation, String description,
171       {bool Function(Set<Choice> choices)? condition})
172       : super(abbreviation, description, condition: condition);
173 }
```

Listing 16: XXXXX, Quelle: Eigenes Listing, Datei: Quellcode/Schritt-4/conditional_form/lib/choices/choices.dart

Teil II

Anhang

Teil III

Implementierung

A Schritt 4 Anhang

B Schritt 5 Anhang

C Schritt 6 Anhang