# Next Steps

* Skript für die Datenbank ausliefern für Patrick
* 🡪Datenbank ins Git exportieren
* UML
  + privaten und Methoden sind unwichtig, Fokus aufs Große und Ganze
  + nicht jedes einzelne Attribut muss dargestellt werden
  + deutliche Darstellung der Verbindungen
* ERD
* Gruppen als Objekte
* Lambda ausimplementieren 🡪 Oracle Dokumentation
* deprecated Methods durch das aktuelle goTo ersetzen
* Testressourcen sollen nicht mit ausgeliefert werden 🡪 Build Management Tool – Maven
* JUnit Klasse fehlt bei der Auslieferung
* Exceptions Handling ausbessern 🡪 alle exceptions auf die GUI integrieren
  + Unbedingt Exception wenn die Datenbank nicht angeschaltet ist
  + wenn exception egal ist, dann umbennenen zu ignore
* VariablenNamen lang und ausführlich, eindeutige Namen (z.B. zurück-Button)
  + DATABASE CONNECTION VARIABLEN
* nicht so viele Strings 🡪 Abändern für die mögliche Internationalisierung 🡪 Scope Doku
  + Internationalisierung auf oracle
* Unit tests: nur einen assert pro Methode, nicht mehrere
* mehr erklären, was genutzt wird und warum, paar Zeilen dazu schreiben um es patrick zu erklären, auch was gut was schlecht daran ist
* trennung der Klasse Screens und Funktionen mithilfe von Interfaces 🡪 UML
* keine static Variablen nutzen 🡪 Doku
* nicht alles public lassen
* ResultSet 🡪 Benutzer soll das nicht nutzen, nur die Daten die er braucht, keine Abhängigkeit, man soll nicht alles aus der Datenbank ziehen
* Datenbank umbennen in dao 🡪 Schnittstelle schaffen
* englische Begriffe
* AdminFunctions umbennen
* Parameter Relation umbennen
* Karte Bild einfügen
* Controller bauen, damit die Screens austauschbar sind -> UML
* Englische Begriffe in den Klassen

<https://github.com/wurstbox/ibait_2020_mvc>

* Patricks beispiel – trennung von ui und internationalisierung ist ok

Prototyp

* Einstieg anders gestalten 🡪 Rolle gestalten, Selektion ob Hund vorhanden oder nicht
* Entscheidungsbaum schon am Anfang mitintegrieren
  + Skizze für Entscheidungsbaum mit in Dokumente aufnehmen
* Prototyp Menü ist überladen
  + einheitliche Gestaltung der Namen
* Filtermöglichkeiten 🡪 In ein Scope Dokument schreiben
* Fehler
  + Verknüpfung AudioGuide-Menu
  + Hundebesitzer, Standorte bei einem selbst
* ERD-Modellierung, um aufzuzeigen, dass über Schnittstelle Daten eingelesen werden
  + Hotelbetten Ausbuchung, Geocaching
* Filterbeispiel auf irgendeinem Screen mit einarbeiten
* Attraktionen zu Übernachtungsmöglichkeiten umbennen (den Benutzer nicht verwirren)
  + 🡪 sowas auch mit in unsere Scope Dokumentation mit aufnehmen
* Feedback von Britta mit in Scope Dokumentation mit aufnehmen

\*\*\*\* = Veronika

\*\*\*\* = Philipp

\*\*\*\* = Nicolas

\*\*\*\* = Lukas