GUI Style-Guide: Nutzerzentrierte Konzeptentwicklung autonomer Fahrzeuge

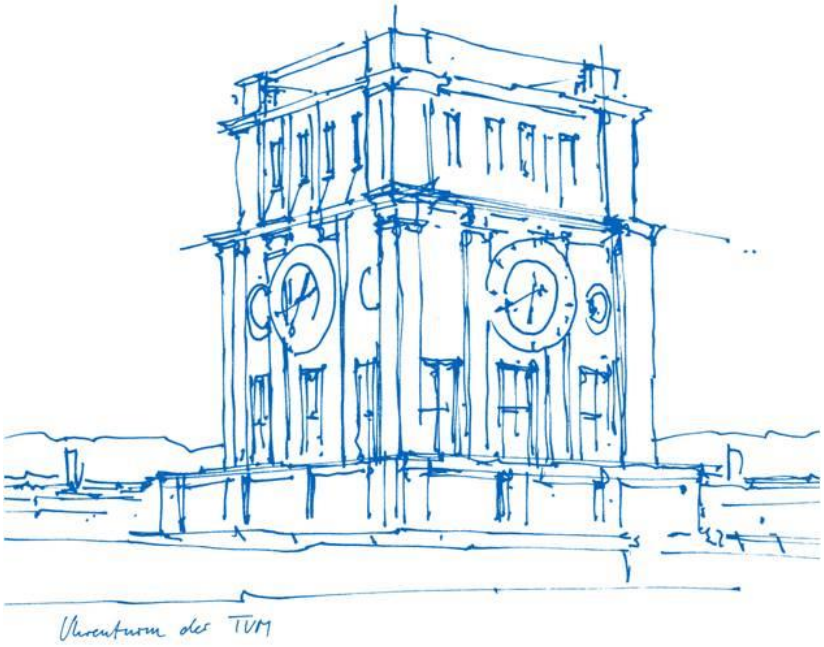
Ferdinand Schockenhoff, David Fischer

Technische Universität München

Fakultät für Maschinenwesen

Lehrstuhl für Fahrzeugtechnik

Garching, 05. August 2021



Inhaltsverzeichnis

1. Prinzipien

- 10 Heuristiken nach Nielsen
- Meta-Prinzipien der Visual Usability

Heuristic 1: „Visibility of system status“

Definition: „The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.“

Tipps:

- Communicate clearly to users what the system’s state is - no action with consequences to users should be taken without informing them
- Present feedback to the user as quickly as possible
- Build trust through open and continuous communication

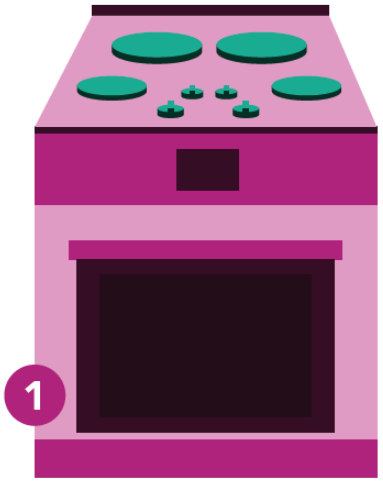
Quelle

Heuristic 2: „Match between system and the real world“

Definition: „The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.”

Tipps:

- Ensure users can understand meaning without having to go look up a word’s definition.
- Never assume your understanding of words or concepts will match those of your users.
- User research will help you uncover your users' familiar terminology, as well as their mental models around important concepts.



Quelle

Heuristic 3: „User control and freedom“

Definition: „Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.”

Tipps:

- Support Undo and Redo.
- Show a clear way to exit the current interaction, like a Cancel button.
- Make sure the exit is clearly labeled and discoverable.

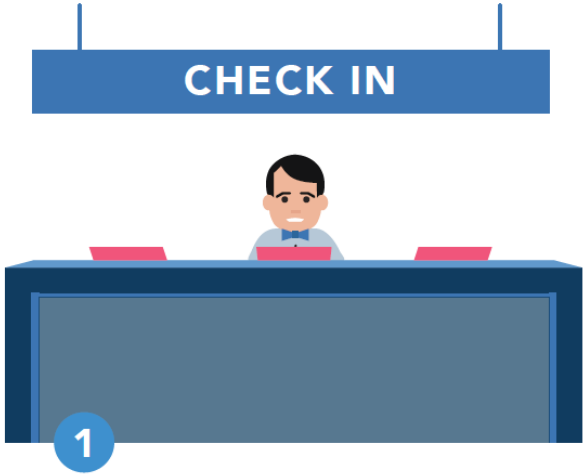
Quelle

Heuristic 4: „Consistency and standards“

Definition: „Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions.”

Tipps:

- Improve learnability by maintaining both types of consistency: internal and external.
- Maintain consistency within a single product or a family of products (internal consistency).
- Follow established industry conventions (external consistency).



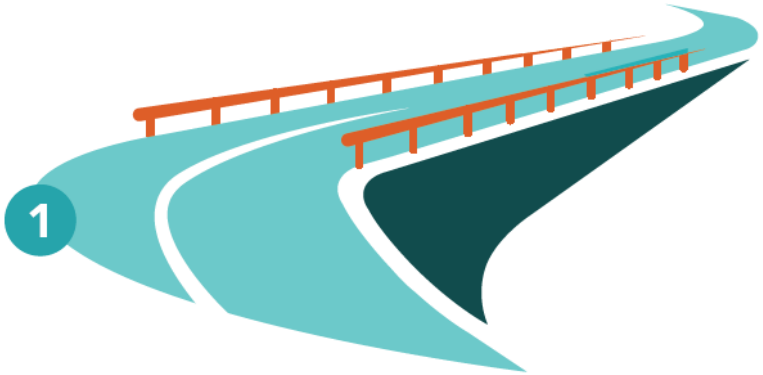
Quelle

Heuristic 5: „Error prevention“

Definition: „Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions, or check for them and present users with a confirmation option before they commit to the action.

Tipps:

- Prioritize your effort: Prevent high-cost errors first, then little frustrations.
- Avoid slips by providing helpful constraints and good defaults.
- Prevent mistakes by removing memory burdens, supporting undo, and warning your users.



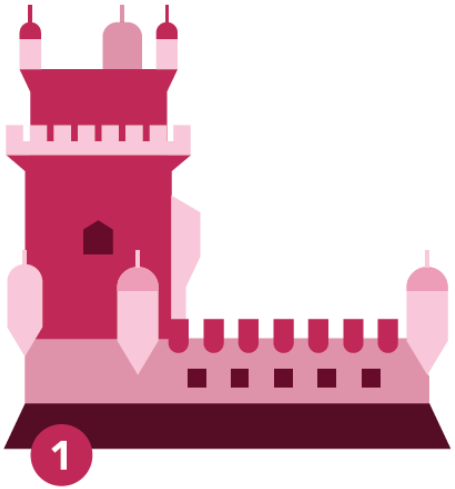
Quelle

Heuristic 6: „Recognition rather than recall“

Definition: „Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.”

Tipps:

- Let people recognize information in the interface, rather than having to remember (“recall”) it.
- Offer help in context, instead of giving users a long tutorial to memorize.
- Reduce the information that users have to remember.



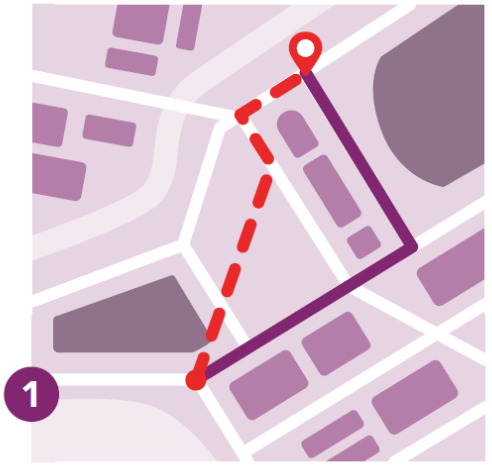
Quelle

Heuristic 7: „Flexibility and efficiency of use“

Definition: „Shortcuts — hidden from novice users — may speed up the interaction for the expert user such that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Tipps:

- Provide accelerators like keyboard shortcuts and touch gestures.
- Provide personalization by tailoring content and functionality for individual users.
- Allow for customization, so users can make selections about how they want the product to work.



Quelle

Heuristic 8: „Aesthetic and minimalist design“

Definition: „Interfaces should not contain information which is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.”

Tipps:

- Keep the content and visual design of UI focus on the essentials.
- Don't let unnecessary elements distract users from the information they really need.
- Prioritize the content and features to support primary goals.

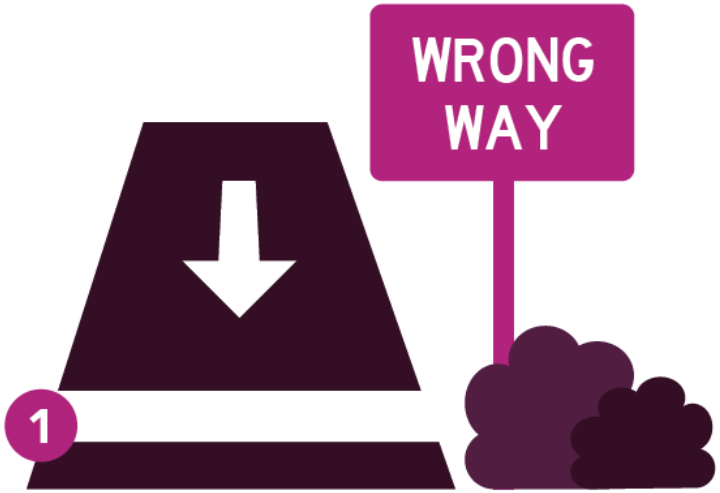
Quelle

Heuristic 9: „Help users recognize, diagnose, and recover from errors“

Definition: „Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.”

Tipps:

- Use traditional error message visuals, like bold, red text.
- Tell users what went wrong in language they will understand — avoid technical jargon.
- Offer users a solution, like a shortcut that can solve the error immediately.



Quelle

Heuristic 10: „Help and documentation“

Definition: „It’s best if the system doesn’t need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.”

Tipps:

- Ensure that the help documentation is easy to search.
- Whenever possible, present the documentation in context right at the moment that the user requires it.
- List concrete steps to be carried out.

Quelle

Meta-Prinzip: Konsistenz

Auf innere und äußere Konsistenz folgender Bereiche achten:

- **Layout**: Wenn Anzeigen dieselben Informationen teilen, sollten diese immer an derselben Stelle erscheinen. Außerdem sollten Elemente die sich aufeinander beziehen immer denselben räumlichen Abstand zueinander besitzen
- **Typographie**: Für Überschriften und Texte sollten Schriftgröße und Schriftart festgelegt sein, ähnlich zu einer Hierarchie um Inkonsistenzen vorzubeugen
- **Farbe**: Über eine definierte Farbpalette kann dem Benutzer bei der Zuordnung von gleichen Elementen geholfen werden
- **Bildliche Darstellung**: In diese Gruppe gehören alle Elemente die nicht den textlichen Elementen zugeordnet werden. Auch hier sollte ein einheitlicher Style der bildlichen Darstellung gewählt werden
- **Steuerelemente**: Durch eine konsistente Verwendung der gleichen Steuerelemente, werden keine neuen Lernprozesse beim Benutzer beansprucht

Meta-Prinzip: Hierarchie

Beispiele zur Erzeugung von Kontrasten:

- **Position**
  - Platzierung und Nähe zum Rand: Elemente am Rand des Bildes werden als weniger wichtig interpretiert als bei zentraler Platzierung.
  - Platzierung und Nähe zu anderen Elementen: Befinden sich viele Elemente dicht zusammen, werden diese als Gruppe identifiziert und können sich so von isolierten Elementen abheben.
  - Verhalten des menschlichen Auges: Aus Studien zum „Eye-Tracking“ ergibt sich eine erste Fokussierung des Benutzers auf die obere linke Ecke.
  - Verschachtelungen: Das Zusammenführen von Elementen in ein größeres Element lässt nicht verschachtelte Elemente vorkommen, als wären diese auf einer höheren Ebene.
  - Überlappungen: Eine Platzierung eines Elements über ein anderes stellt eine einfache Form des Kontrastes dar.
- **Visuelle Handhabung**
  - Größe: Je größer ein Element ist, desto wichtiger erscheint es.
  - Farbe: Bestimmte Signalfarben, wie rot, orange oder gelb stechen mehr hervor als blaue oder grüne Farbtöne.
  - Ornamente: Durch die Einbettung eines Elements in einer dekorativen Umgebung können Kontraste erzeugt werden.
  - Finish: 3D-Effekte für Steuerelemente können die Aufmerksamkeit erhöhen, im Gegensatz zur einfarbigen Gestaltung.

Meta-Prinzip: Charakter

- GUI hinterlässt Eindruck auf den Benutzer
- Positiven Widererkennungswert schaffen
- **Prägende Elemente des Charakters:**
  - Benutzer -> Gemachte Erfahrungen und Erwartungen
  - Entwicklungsorganisation -> Style-Guides und frühere Entwicklungen
  - Stakeholder -> Termine und Ziele
  - Designer/Entwickler -> Voreinstellungen und Präferenzen

2. Grundlagen und Grundelemente

- Farbpalette
- Schriftarten und –größen
- Sprache
- Eingabeelemente
- Buttons
- Panels
- Labels, Text Edit Fields, Text Area
- Bilder

Farbpalette

Farben zur Verwendung sollen aus den TUM-Colors entnommen werden. Zuerst sollen Farben aus der primären und sekundären Farbpalette verwendet werden. Bei Bedarf sind Akzentfarben erlaubt. Die erweiterte Farbpalette ist nur in Ausnahmefällen erlaubt.

- Primär

Blue 0,0.4,0.74	Black 0,0,0	White 1,1,1

- Sekundär

LightBlue 0,0.32,0.58	DarkBlue 0,0.2,0.35	LightGrey 0.85,0.85,0.86	Grey 0.61,0.62,0.62	DarkGrey 0.36,0.36,0.35

- Akzentfarben

Orange 0.89,0.45,0.13	Green 0.64,0.68,0	LightBlue 0.6,0.78,0.92	DarkBlue 0.39,0.63,0.78	Beige 0.85,0.84,0.80

- Erweiterte Farben

E1 0.41,0.03,0.35	E2 0.06,0.11,0.04	E3 0.2,0.47,0.54	E4 0,0.49,0.19	E5 0.40,0.60,0.11
E6 1,0.86,0	E7 0.98,0.73,0.0	E8 0.84,0.3,0.07	E9 0.77,0.03,0.11	E10 0.61,0.05,0.09

Schriftarten und -größen

- **Überschrift 1: Arial 24 fett**
- **Überschrift 2: Arial 16 fett**

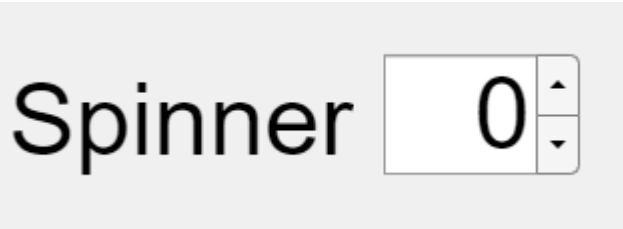
- Textschrift 1: Arial 16
- Textschrift 2: Arial 12

Sprache

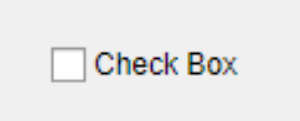
- Verwendete Sprache: Englisch (US)
- Alles wird klein geschrieben bis auf Eigennamen
- Eigennamen:
  - Namen der GUIs
  - Fachbegriffe: Derivative, Vehicle Concept, Customer-relevant Properties; Technical Properties, User, Scenario, Vehicle Provision, User Fulfillment

Eingabeelemente

Für Parametereingaben sind „Spinner“ zu verwenden



Für Ja/Nein Abfragen sind Checkboxen zu verwenden



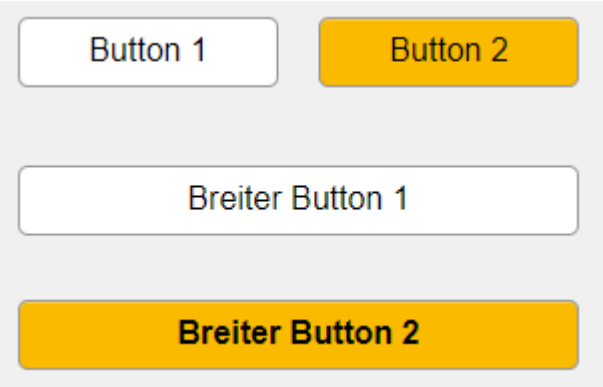
Zur Auswahl von vorgegebenen Einstellungsmöglichkeiten Dropdown verwenden

⇒ Größen sind an das Menü und die Schriftgrößen anzupassen

⇒ Wenn möglich die Parameter eingrenzen

Buttons

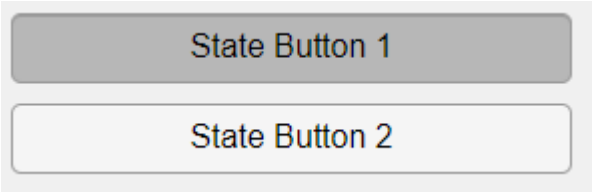
Push-Buttons -> Verwendung bei merhmaliger Ausführung einer Funktion



- Hintergrundfarbe ist weiß (1,1,1); bei Hervorhebung wird erweiterte Farbe 7 (0.98,0.73,0.00) verwendet
- Normale Größe: 120px35p
- Breite Größe: 280x35p
- In Ausnahmefällen ist die Breite und Höhe anpassbar
- Schriftarten entsprechend der Textschriftarten

Buttons

State-Buttons -> Ausführen der Funktion einmalig, danach wird Status gehalten



- Größe, Farbe, Schriftart -> wie bei Push-Button

Panels

Haupt Panel

- Hintergrundfarbe weiß (1,1,1) oder grau (0.85,0.85,0.86)
- Überschrift zentriert und in Überschrift 1 (fett, Arial 24)
- BorderType: None

Panels

Unter-Panel

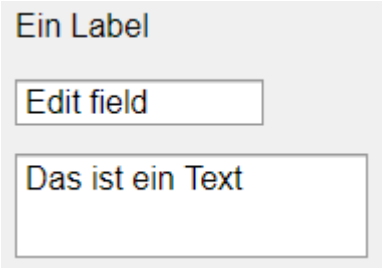
- Hintergrundfarbe weiß (1,1,1) oder grau (0.85,0.85,0.86) -> passt sich der Farbe des Haupt-Panels an!
- Überschrift zentriert und in Überschrift 2 (fett, Arial 16)
- BorderType: None

Labels, Text Edit Fields, Text Areas

Label
- Verwendung als Beschreibung oder große Überschriften
- Größe passt sich der Schriftart des zu beschreibenden Elements an
- Farbe so wählen dass Kontrast gegeben für Lesbarkeit

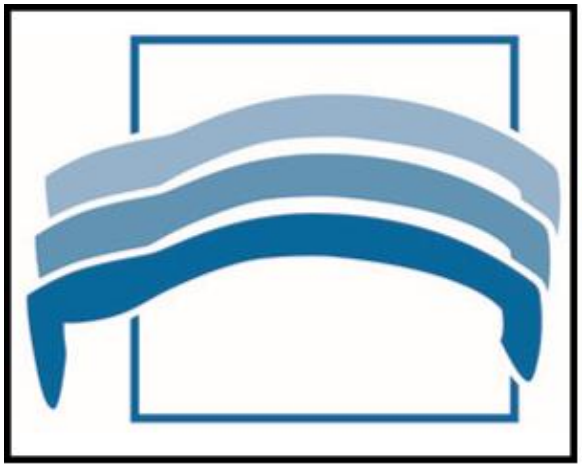
Text Edit Fields
- Verwendung bei veränderlichen Beschreibungen
- Signalisieren einen veränderlichen Wert
- Größe und Schriftart orientiert sich an dem zu beschreibenden Element
- Hintergrundfarbe ist weiß (1,1,1)

Text Areas
- Verwendung nur bei größeren Beschreibungstexten
- Variable Größe
- Hintergrundfarbe ist weiß (1,1,1)



Bilder

- Bilder in möglichst hoher Auflösung
- Schwarzer Rand als Rahmen



3. Komponenten

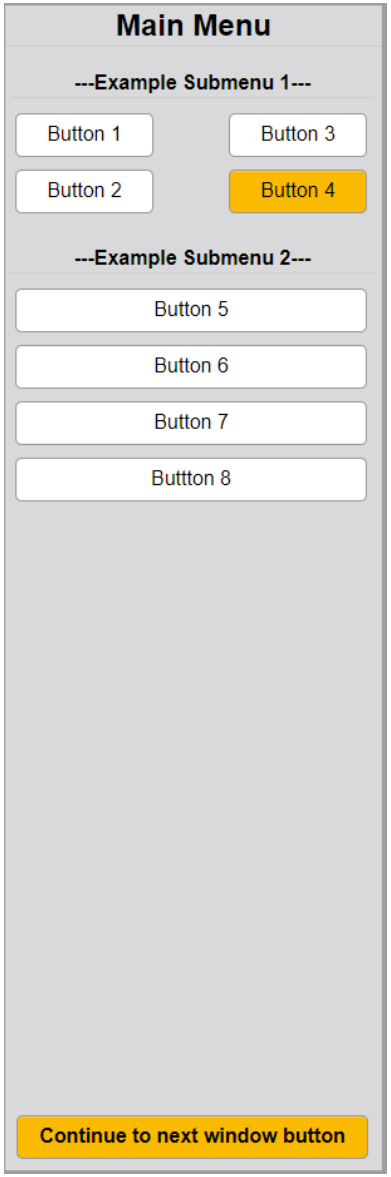
- Namensbanner
- Hauptmenü
- Hauptanzeige
- Untermenü im Hauptmenü
- Manuelle Auswahl
- Parameter Panel
- Spinnendiagramme

Namensbanner

- Anzeige des GUI-Namen und TUM Logo
- Größe und Positionierung: (0,940,1921,71)
- Um Abstand zu linkem Rand zu generieren die ersten vier Zeichen leer lasse
- Hintergrundfarbe: Sekundär – Dark Blue (0.00,0.20,0.35)
- Schriftart: Arial 32 weiß

Hauptmenü

- Beinhaltet alle Funktionen der GUI, die sich auf zwei oder mehrere Ansichten beziehen
- Kann in Untermenüs gegliedert werden
- Größe: (5,5,300,930)
- Hintergrundfarbe: Sekundär – Light Grey (0.85,0.85,0.86)
- Schriftart: Überschrift 1 – Arial 24 fett
- BorderType: None



Hauptanzeige

- Darstellung von zusammenhängenden Inhalten
- Kann Unter-Panels zur Strukturierung beinhalten
- Für mehrere Ansichten werden Hauptanzeigen übereinander gelegt und per Back-End angesteuert
- Größe: (310,5,1605,930)
- Hintergrundfarbe: weiß (1,1,1)
- Schriftart: Überschrift 1 – Arial 24 fett
- BorderType: None

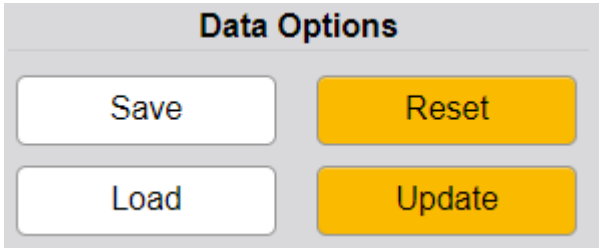
Untermenü im Hauptmenü

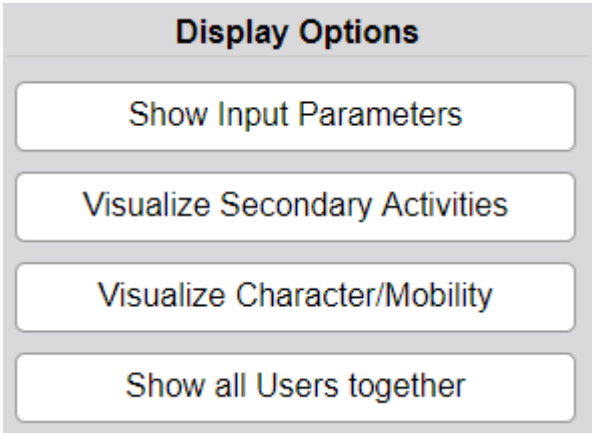
- Gruppierung von Funktionen aus dem selben Funktionsbereich
- Größe orientiert sich an Breite des Hauptmenüs und Anzahl der gruppierten Elemente
- Breite: 290p
- Höhe nach Formel: (n = Anzahl der Elemente in vertikaler Richtung)

$$H = 35p + 15p + n * 35p + (n - 1) * 10p$$

- Hintergrundfarbe: Sekundär – Light Grey (0.85,0.85,0.86)
- Schriftart: Überschrift 2 – Arial 16 fett
- BorderType: None

Beispiele:





Manuelle Auswahl

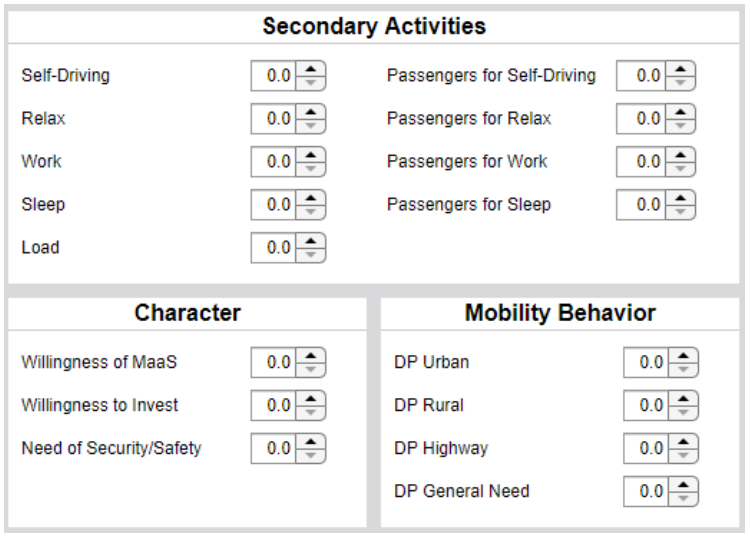
- Bei der Auswahlmöglichkeit mehrerer Graphen oder zur Visualisierung einer Legende
- Setzt sich aus Unter-Panel mit beschrifteten Lampen, die per Check-Box angeklickt werden können zusammen
- Größe variiert je nach Anzahl der Parameter

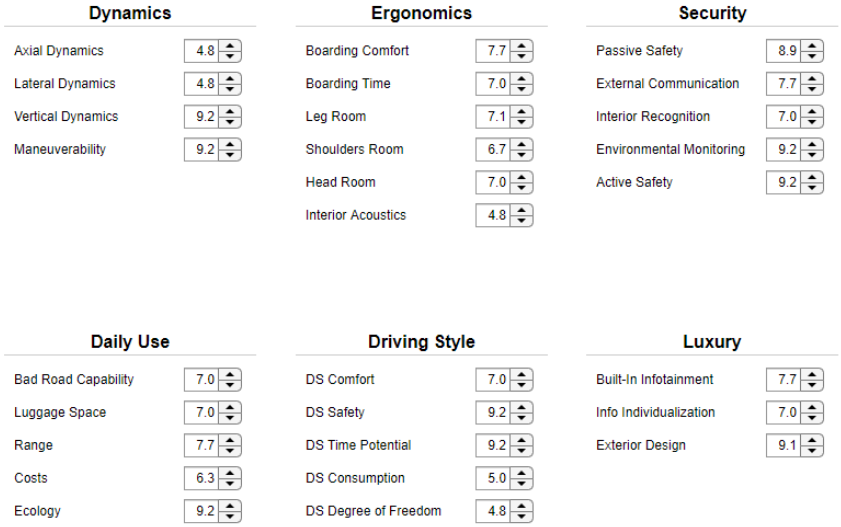
Beispiel:

Parameter Panel

- Zusammenfassung mehrerer Spinner der gleichen Kategorie in einem Unter-Panel
- Größe der Spinner hängt von Beschriftungsbreite ab
- Bei Verwendung mehrerer Kategorien, Spaltenordnung übergeordnet behalten
- Hintergrund wahlweise weiß oder dem übergeordneten Unter-Panel entsprechend

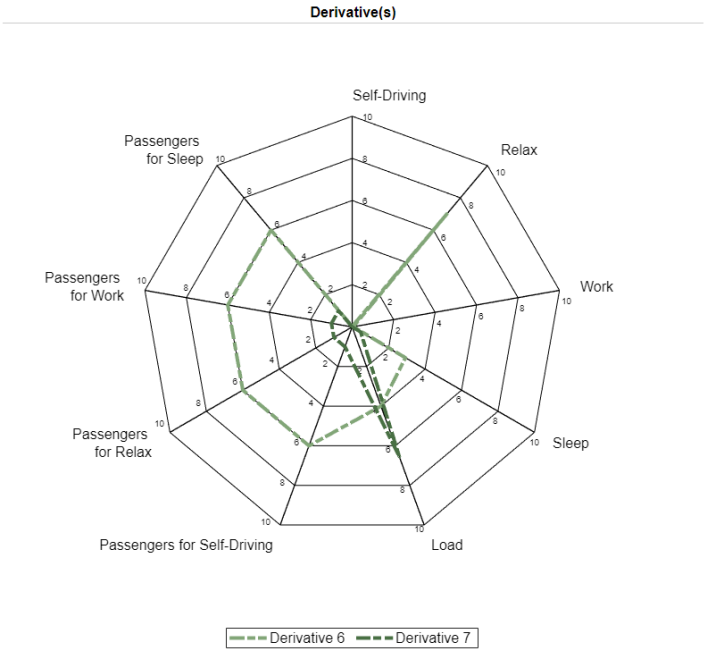
Beispiele:

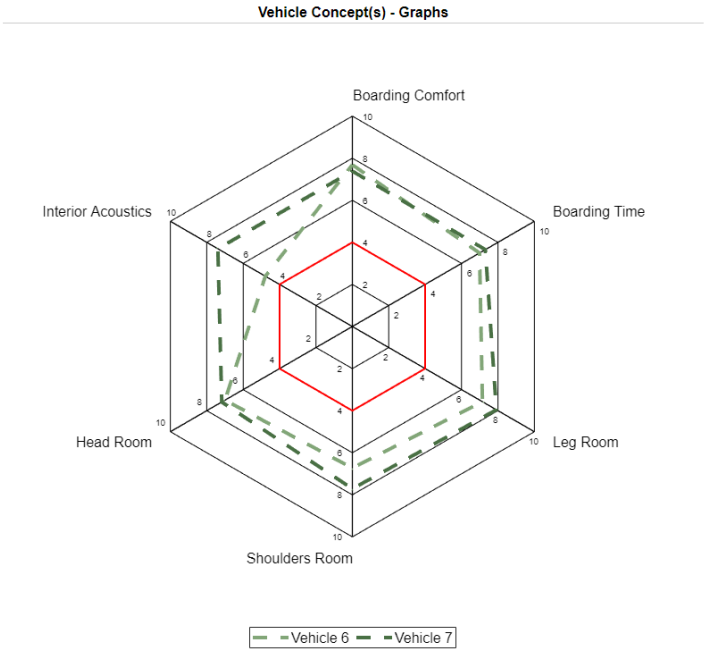




Spinnendiagramme

- Erstellung über die Matlab Funktion „plotSpiderGraph“
- In groß und klein verfügbar
- Vorher in der UIAxes die Achsenfarbe auf Hintergrundfarbe ändern
- Titel entfernen
- Immer in einem Panel platzieren
- Größe und Positionierung hängen von Panelgröße ab





4. Templates

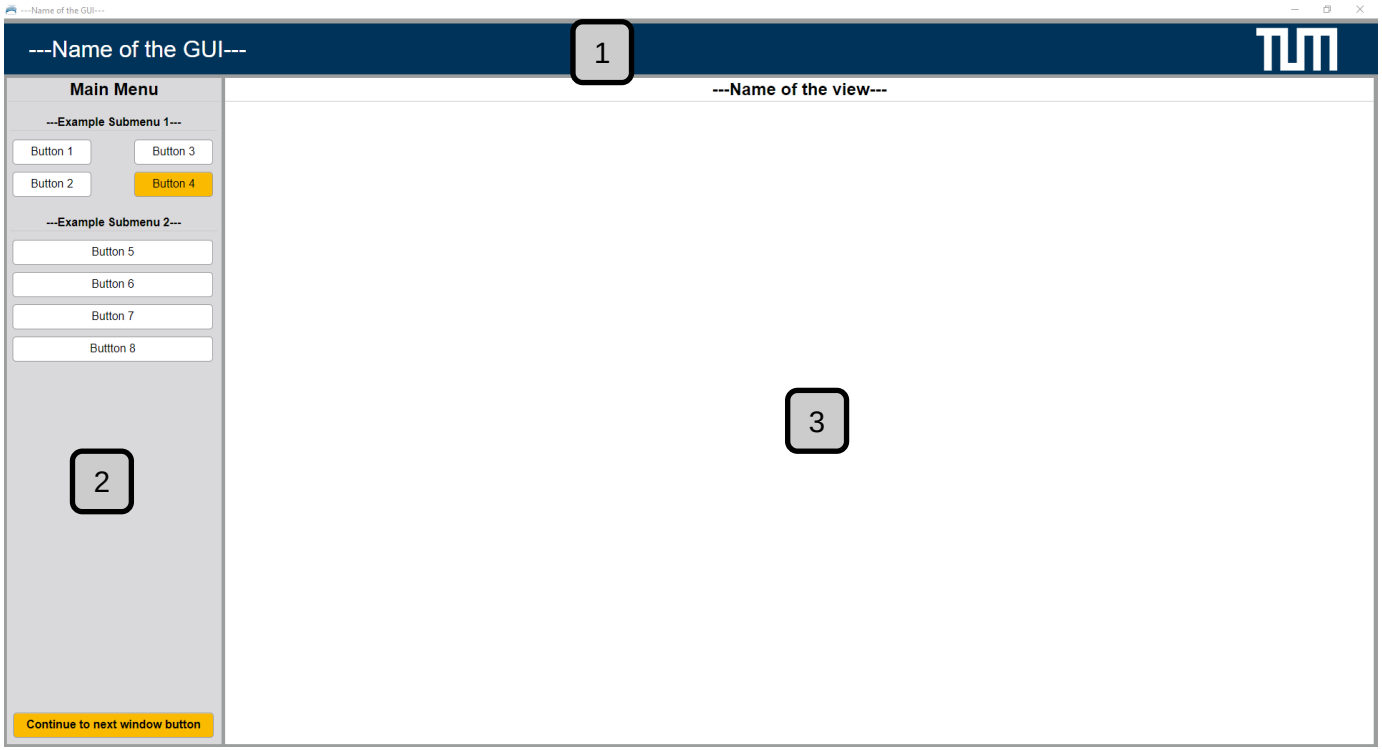
- Hauptbildschirm
- Hauptlayout
- Layout Hauptmenü
- Layout Hauptanzeige 2-Panel
- Layout Hauptanzeige 6-Panel

Hauptbildschirm

- Entspricht der app.UIFigure
- Wird als Referenz für Meldungen verwendet
- Legt die Gesamtgröße der GUI fest
- Auslegung auf 1920x1080p Bildschirmgröße
- Verkürzung der Höhe, damit untere Taskleiste beachtet wird -> 1020p
- Hochrücken um 40p damit richtige Platzierung
- Hintergrundfarbe ist sekundäres grau (0.61,0.62,0.62)
- Möglichkeit des Scrollens
- AutoResizeChildren aus
- Name und Icon der GUI eintragen
  - Icon ist das FTM Logo

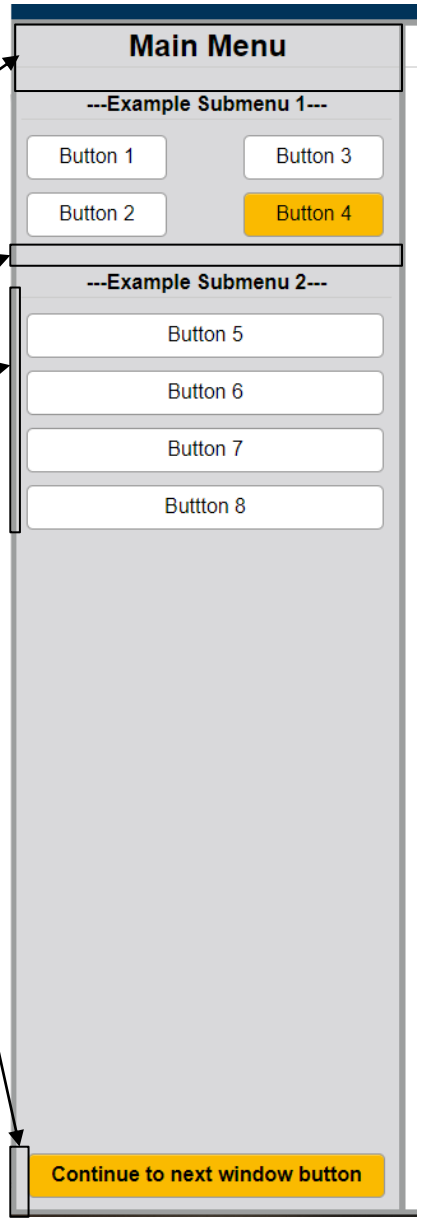
Hauptlayout

- Platzierung des Namensbanner oben (1)
- Hauptmenü auf der linken Seite (2)
- Hauptanzeige füllt den Rest aus (3)
- Abstände zwischen den drei Komponenten sind 5p
- Abstände zum Rand auch 5p (außer Namensbanner Breite)



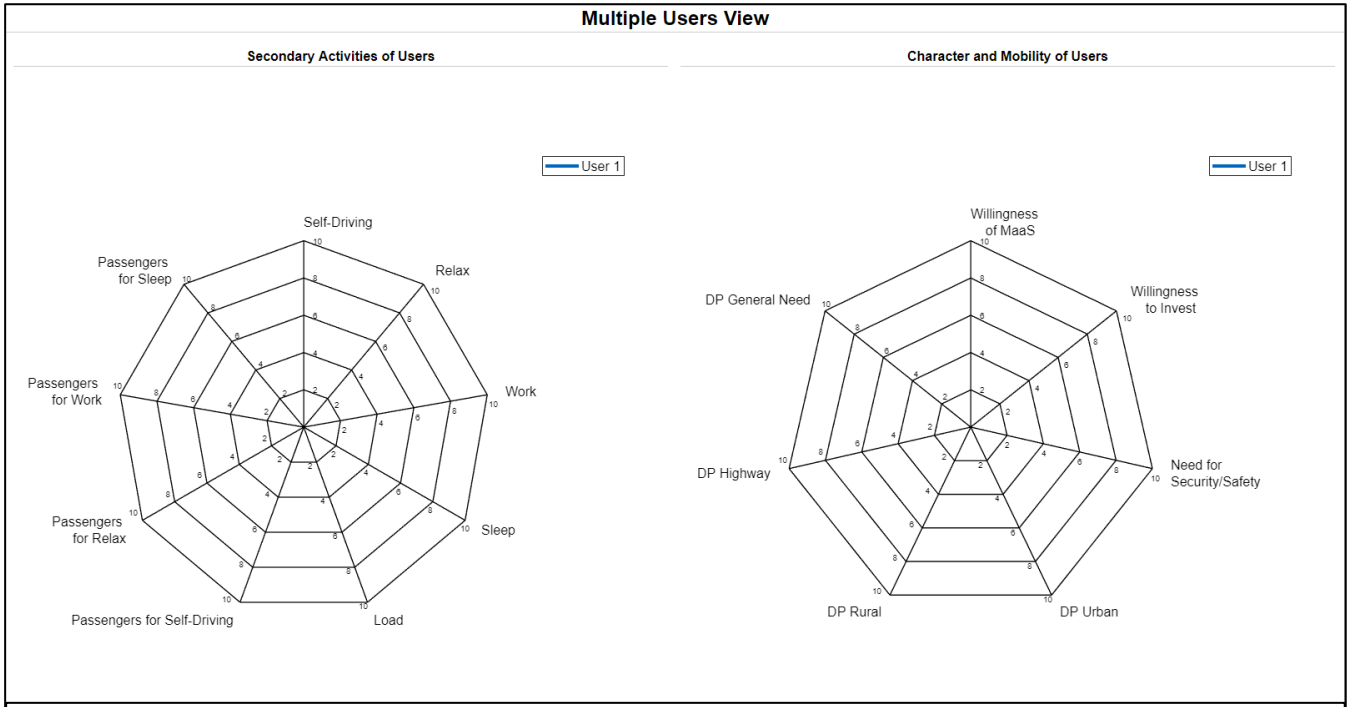
Layout Hauptmenü

- Inhalte werden von oben nach unten gefüllt
- Platz zur Überschrift des Haupt-Panels 50p
- Zwischen den Komponenten und Elementen sind Abstände von mindestens 10p
- Randabstand seitlich eines Unter-Panels 5p
- Randabstand seitlich eines Standardelements 10p
- Optional: Überleitung zur nächsten GUI ganz unten platzieren (x=10p, y=10p)



Layout Hauptanzeige 2-Panel

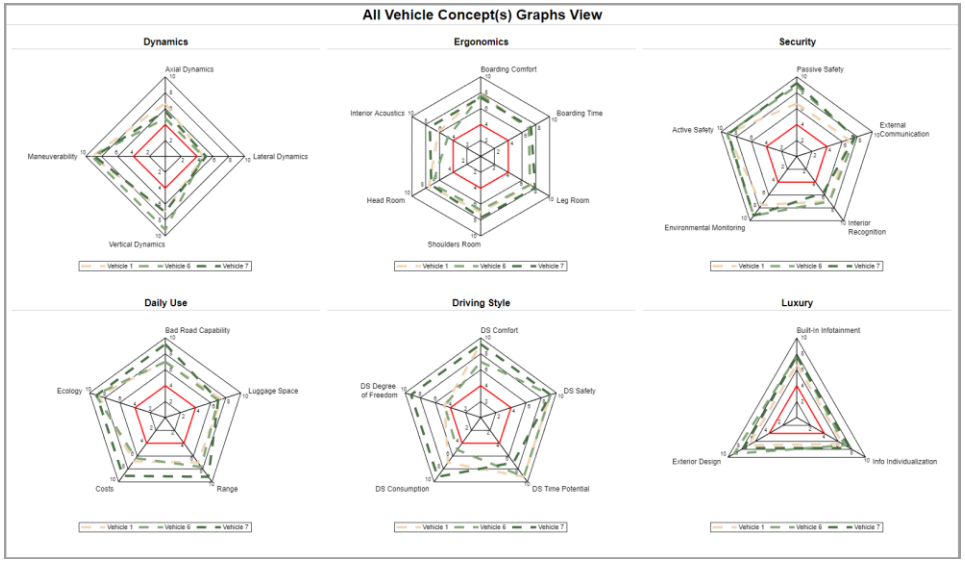
- Platz zur Überschrift des Haupt-Panels 50p
- Abstände zwischen beiden Panels und Rändern 10p
- Größen/Positionierung der Unterpanels: 10,10,785,870 / 810,10,785,870
- Bei Verwendung unterer Leiste mit Höhe 105p: 10,115,785,765 / 805,115,785,765



Layout Hauptanzeige 6-Panel

- Platz zur Überschrift des Haupt-Panels 50p
- Abstände zwischen beiden Panels und Rändern 10p
- Größe/Positionierung Unterpanel 1: 10,450,520,430 (10,500,520,380)
- Größe/Positionierung Unterpanel 2: 542,450,520,430 (542,500,520,380)
- Größe/Positionierung Unterpanel 3: 1075,450,520,430 (1075,500,520,380)
- Größe/Positionierung Unterpanel 4: 10,10,520,430 (10,115,520,380)
- Größe/Positionierung Unterpanel 5: 542,10,520,430 (542,115,520,380)
- Größe/Positionierung Unterpanel 6: 1075,10,520,430 (1075,115,520,380)

In Klammern bedeutet mit Unterleiste 115p



5. Umsetzungen

- App-Launcher
- GUI: User-Centered Mobility Needs
- GUI: Vehicle-Bound Mobility Provision
- GUI: Customer-Relevant Properties

Nutzungsanforderungen

Tabelle 4.9: Nutzungsanforderungen

	Anforderung	Index
Sichtbarkeit des Systemstatus	Benutzer soll immer Bescheid wissen was gerade in der GUI passiert und was an Bedienungen oder Eingaben möglich sind	NA-01
	Benutzer soll immer Bescheid wissen wo er sich gerade befindet	NA-02
	Fortschritt soll für aktive und passive Schritte visualisiert sein	NA-03
	Bei erfolgreicher Fertigstellung von Aufgaben soll Rückmeldung gegeben werden	NA-04
Übereinstimmung System und Realität	Jeglicher Inhalt muss für die Benutzergruppen verständlich sein	NA-05
	Natürliche und logische Reihenfolgen sollen befolgt werden	NA-06
Benutzerkontrolle und Freiheit	Möglichkeit des rückgängig Machens von Aktionen	NA-07
	Benutzer soll die Möglichkeit haben aus kritische Situationen zu beenden	NA-08
	Benachrichtigungen beim Aktivieren von wichtigen oder kritischen Aktionen	NA-09
Konsistenz und Standards	Konsistenz in Bedeutung, Funktion und Organisation	NA-10
	Einhaltung von Standards in Layout und Sprache	NA-11
Fehlerprävention	Parameter wenn nötig eingrenzen	NA-12
	Bestätigungsaufruf beim Aktivieren von wichtigen oder kritischen Funktionen	NA-13
	Warnung bei der Änderung einer Inhalts, der Ausgaben maßgeblich beeinflusst	NA-14
	Eingabe von Parametern in verschiedenen Formaten unterstützen	NA-15
	Möglichkeit des Zurücksetzens der GUI	NA-16
Wiedererkennen anstatt Erinnern	Alle benötigten Informationen zur Erledigung der Aufgabe sollen sich in der Anzeige befinden	NA-17
Flexibilität und effiziente Nutzung	Aufgaben sollen in ihrer einfachsten Form erfüllbar sein	NA-18
Ästhetisches und minimalistisches Design	Hinterlassen eines guten subjektiven Eindrucks	NA-19
	Die GUI sollte klar strukturiert sein	NA-20
	Eine Informations- und Reizüberflutung soll verhindert werden	NA-21
	Schriftgrößen sollen ausreichend gestalten sein	NA-22
	Zur farblichen Abgrenzung sollen Kontraste ausreichend gestaltet werden	NA-23
Hilfe und Dokumentation	Verwendung von Tool Tipps zur Hilfestellung	NA-24

App-Launcher

Kontext:

- Übergeordnete Start-GUI, die einen Überblick über den Gesamtprozess gibt
- Außerdem lassen sich alle anderen GUI von hier aus starten
- Optional können beim Start Szenarien ausgewählt werden

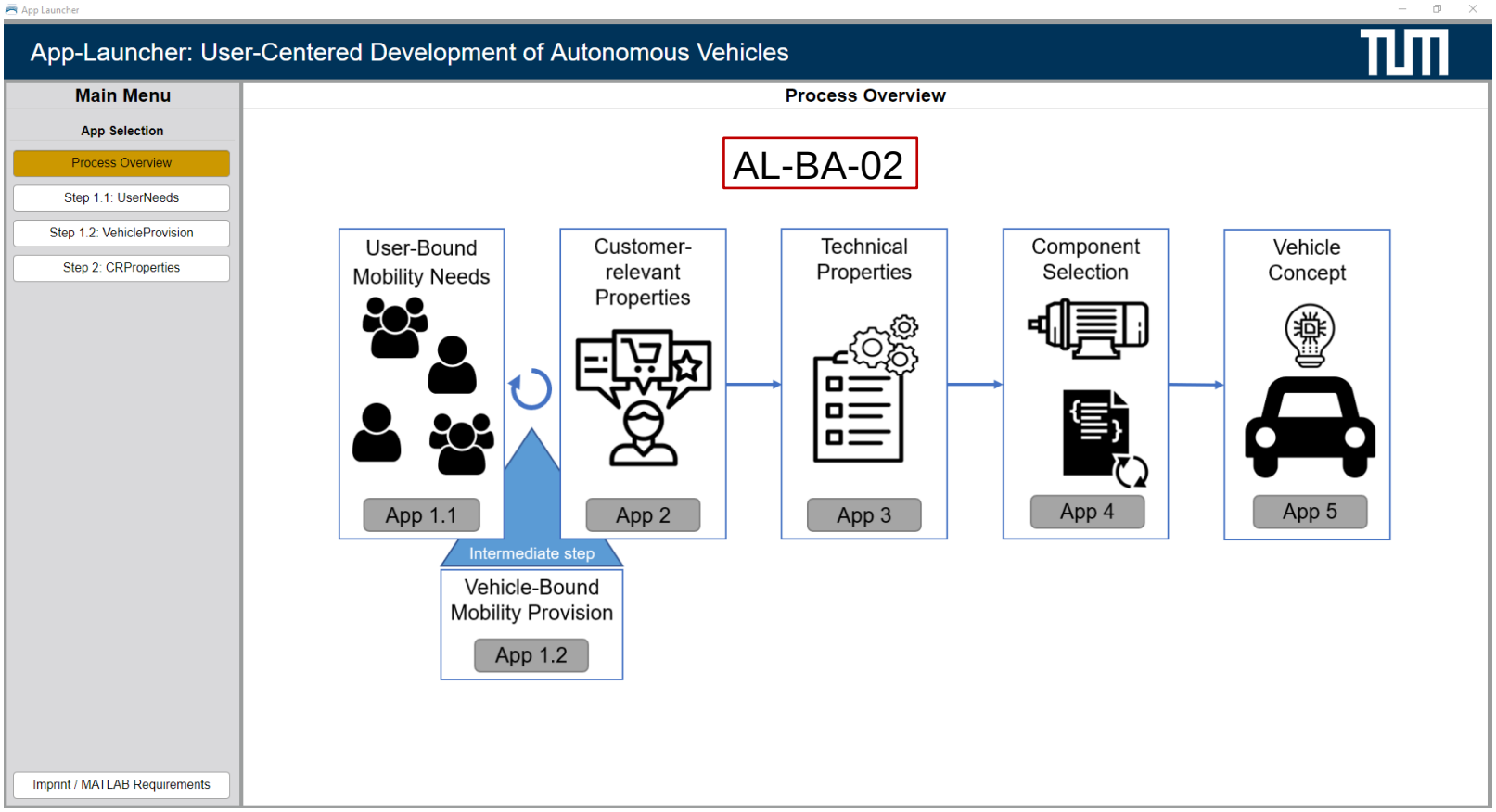
Entwicklungsaufgabe: (siehe Benutzeranforderungen)

Tabelle 4.8: Benutzeranforderungen App-Launcher

GUI	Anforderung	Index
App-Launcher	Starten der GUIs, optional mit Szenario	AL-BA-01
	Übersicht des Entwicklungsprozesses mit Zuordnung der GUIs	AL-BA-02
	Vorschau der GUIs	AL-BA-03
	Beschreibungstext für die GUIs	AL-BA-04
	Anzeige des Impressums und der Programmvoraussetzungen	AL-BA-05

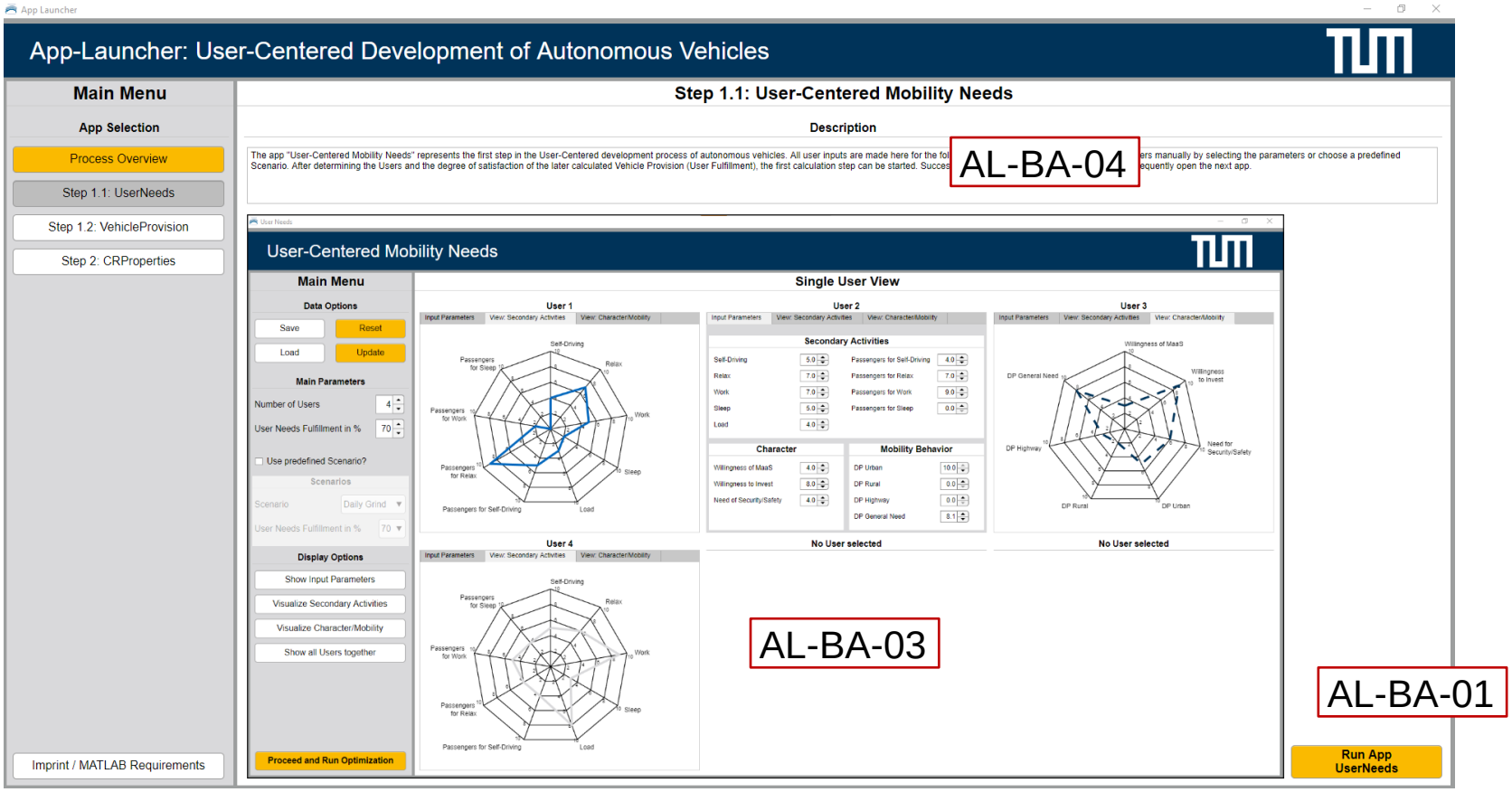
App-Launcher

Anordnungen und Erfüllung Nutzerbedürfnisse:



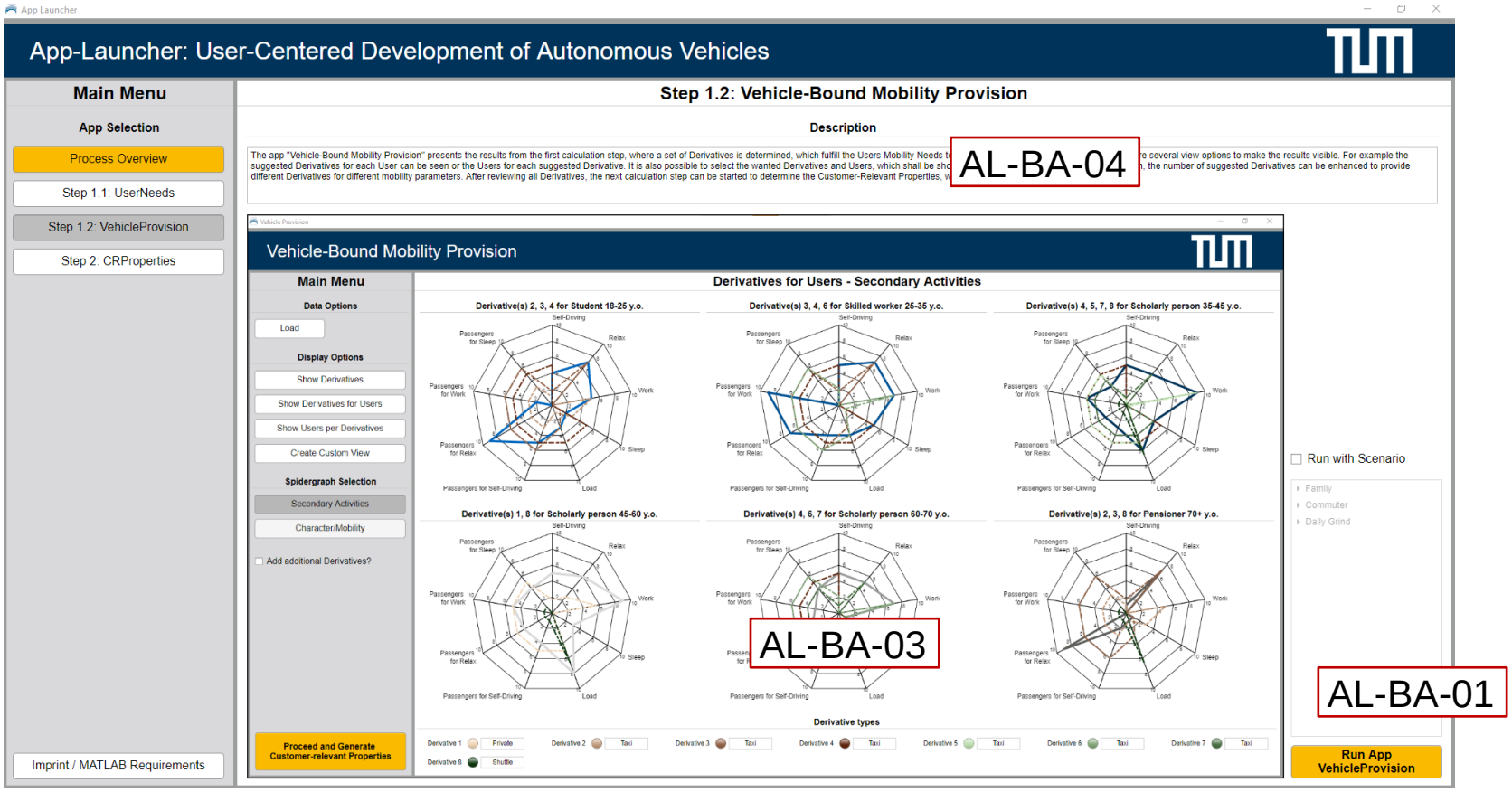
App-Launcher

Anordnungen und Erfüllung Nutzerbedürfnisse:



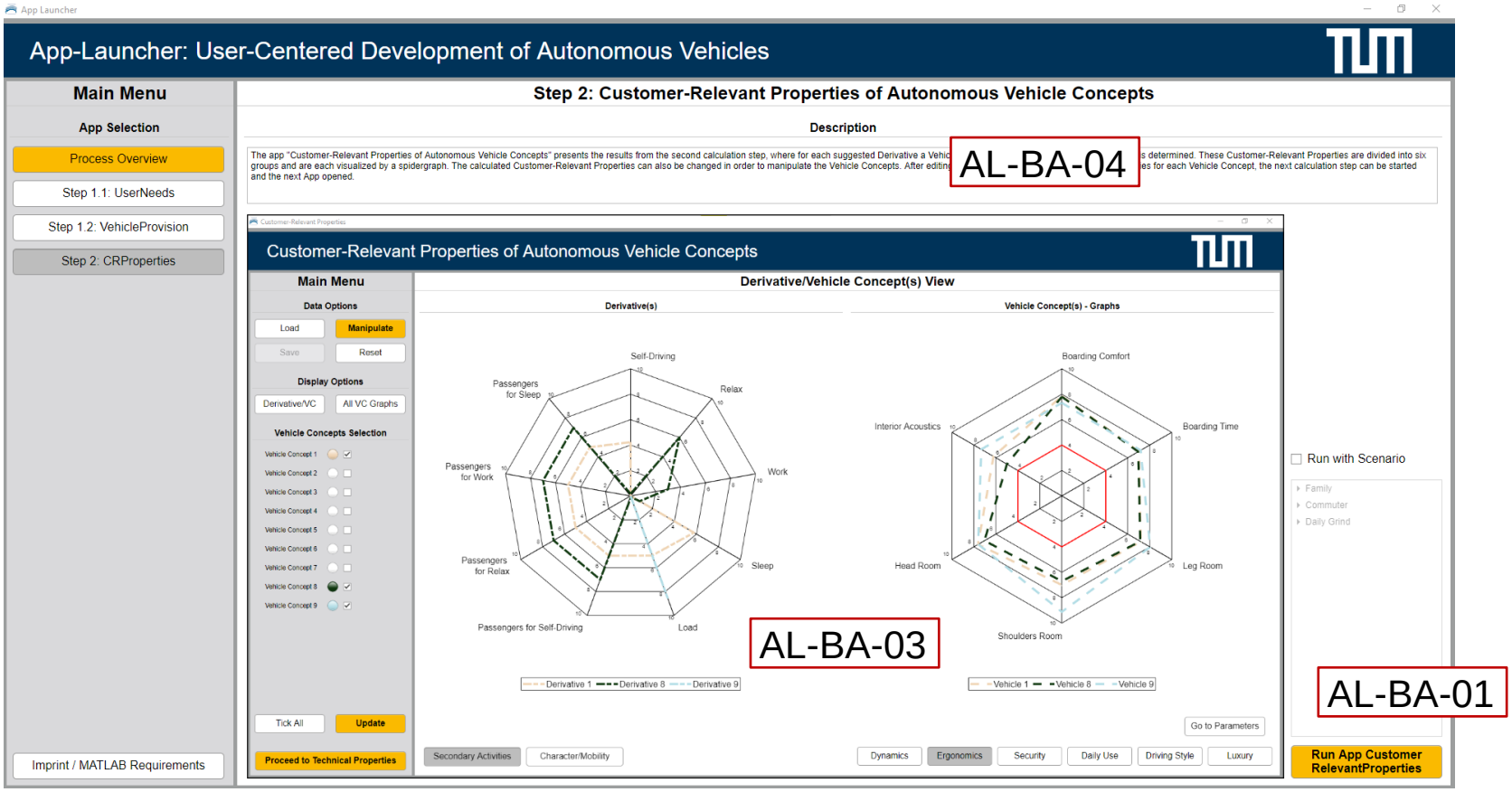
App-Launcher

Anordnungen und Erfüllung Nutzerbedürfnisse:



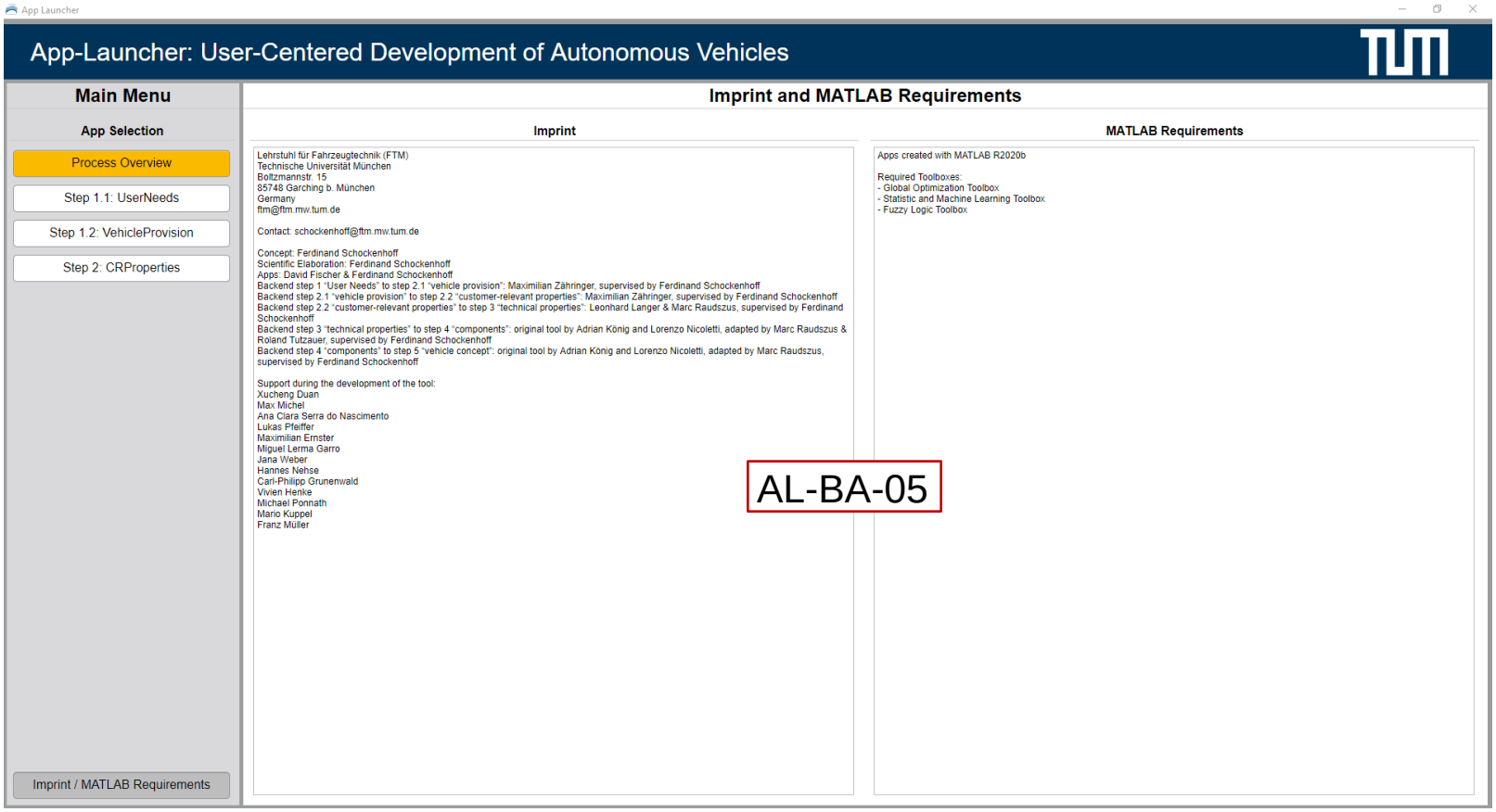
App-Launcher

Anordnungen und Erfüllung Nutzerbedürfnisse:



App-Launcher

Anordnungen und Erfüllung Nutzerbedürfnisse:



Step 1.1: User-Centered Mobility Needs

Kontext:

- Erste GUI, die zur Aufnahme und Abbildung des Mobilitätsbedarfs verwendet wird
- Initiiert ersten Berechnungsschritt zu KWE (Anzahl Derivate)

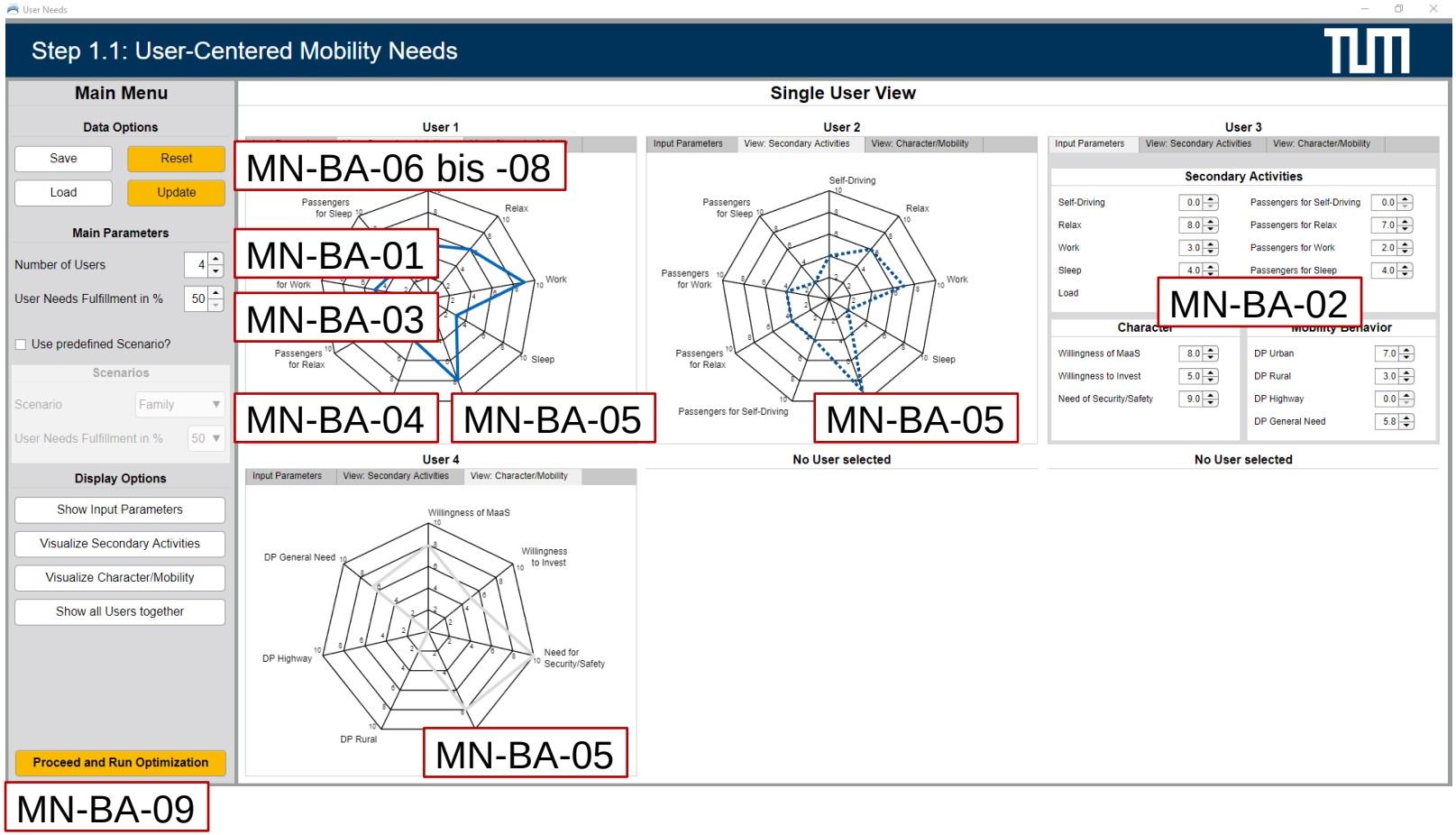
Entwicklungsaufgabe: (siehe Benutzeranforderungen)

Tabelle 4.5: Benutzeranforderungen Mobility Needs

GUI	Anforderung	Index
Mobility Needs	Eingabe von ein bis sechs Nutzern	MN-BA-01
	Eingabe der Mobilitätsparameter	MN-BA-02
	Eingabe des Befriedigungsgrades	MN-BA-03
	Auswahlmöglichkeit von drei vordefinierten Szenarien	MN-BA-04
	Darstellung der Parameter in Spinnengraphen (einzeln und zusammen)	MN-BA-05
	Speicher- und Ladefunktion	MN-BA-06
	Zurücksetzen der Parameter	MN-BA-07
	Manuelles Aktualisieren von Spinnengraphen	MN-BA-08
	Start der Berechnung der Anzahl an Derivaten	MN-BA-09

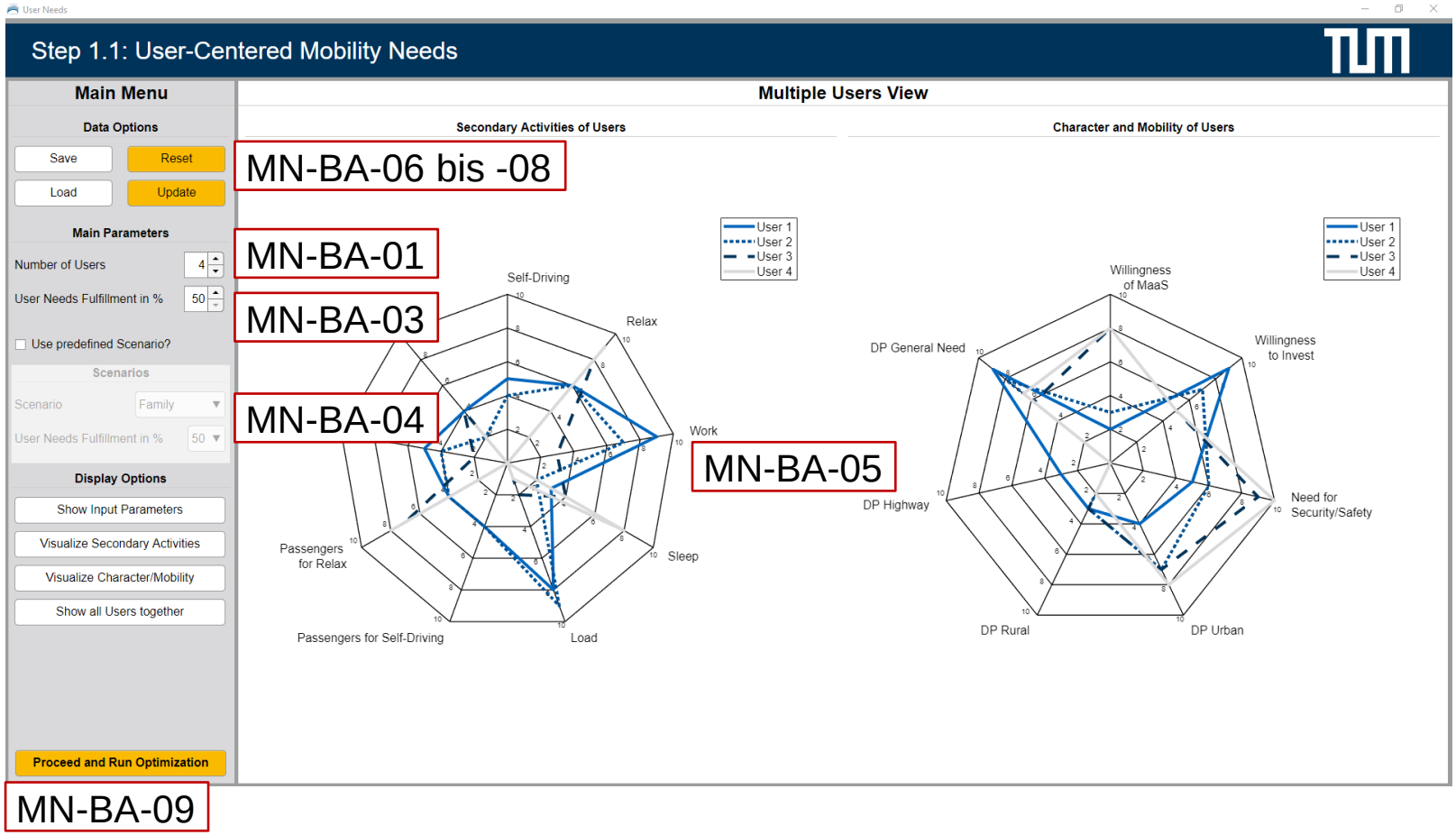
Step 1.1: User-Centered Mobility Needs

Anordnungen und Erfüllung Nutzerbedürfnisse:



Step 1.1: User-Centered Mobility Needs

Anordnungen und Erfüllung Nutzerbedürfnisse:



Step 1.2: Vehicle-Bound Mobility Provision

Kontext:

- Zweite GUI, die zur Repräsentation des ersten Berechnungsschritts verwendet wird
- Darstellung der Derivate mit dem abgedeckten Mobilitätsbedarf
- Initiiert zweiten Berechnungsschritt (KWE der Derivate)

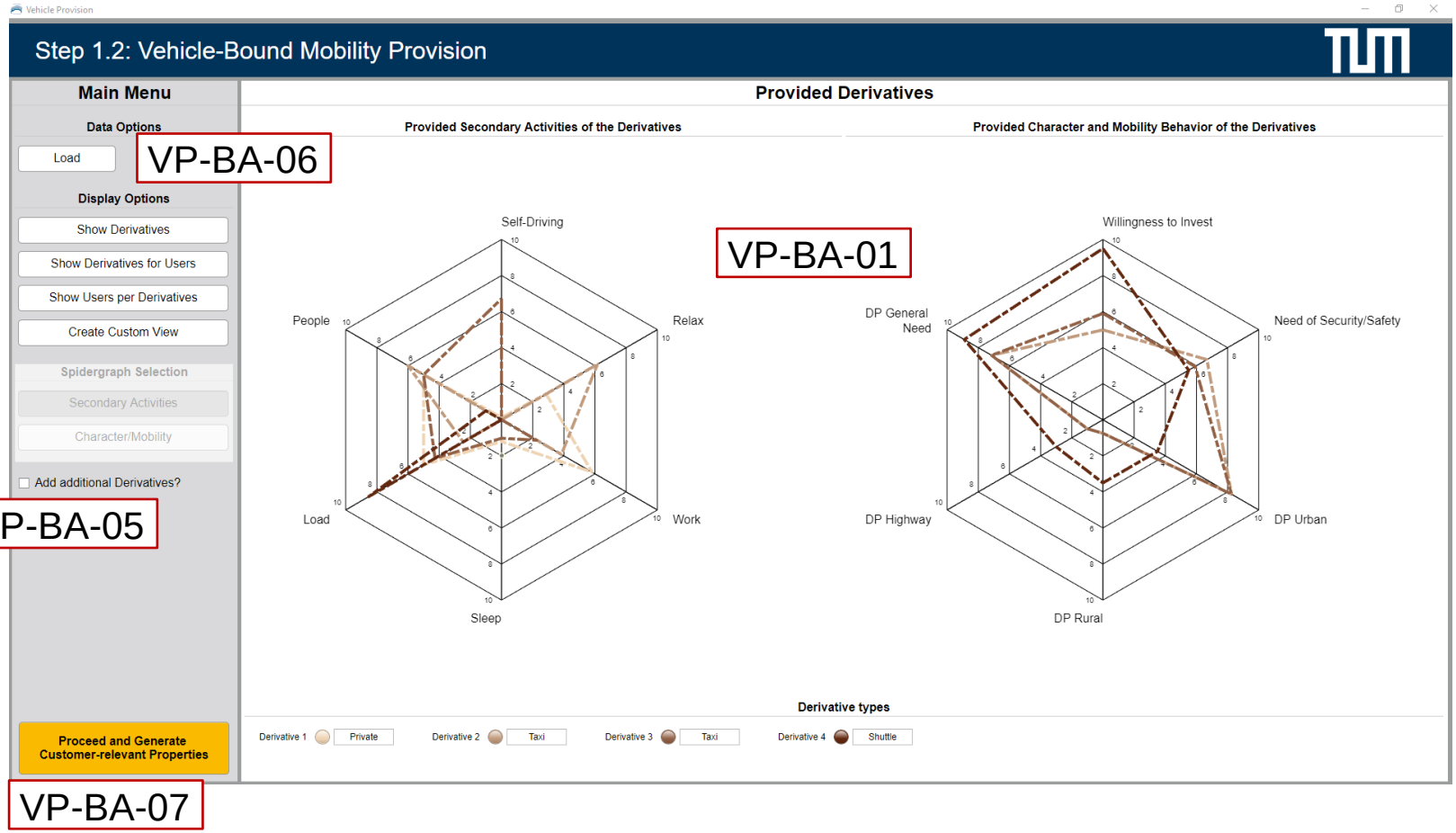
Entwicklungsaufgabe: (siehe Benutzeranforderungen)

Tabelle 4.6: Benutzeranforderungen Vehicle Provision

GUI	Anforderung	Index
Vehicle Provision	Anzeige der Derivate mit abgedeckten Mobilitätsparametern	VP-BA-01
	Darstellung der Derivate und für zugeordnete Nutzer	VP-BA-02
	Darstellung der Nutzer für vorgeschlagene Derivate	VP-BA-03
	Selbstbestimmte Auswahl von Nutzern und Derivaten für Darstellung	VP-BA-04
	Möglichkeit zusätzliche Derivate hinzuzufügen	VP-BA-05
	Laden von gespeicherten Berechnungsergebnissen	VP-BA-06
	Start der Berechnung der kundenwerten Eigenschaften	VP-BA-07

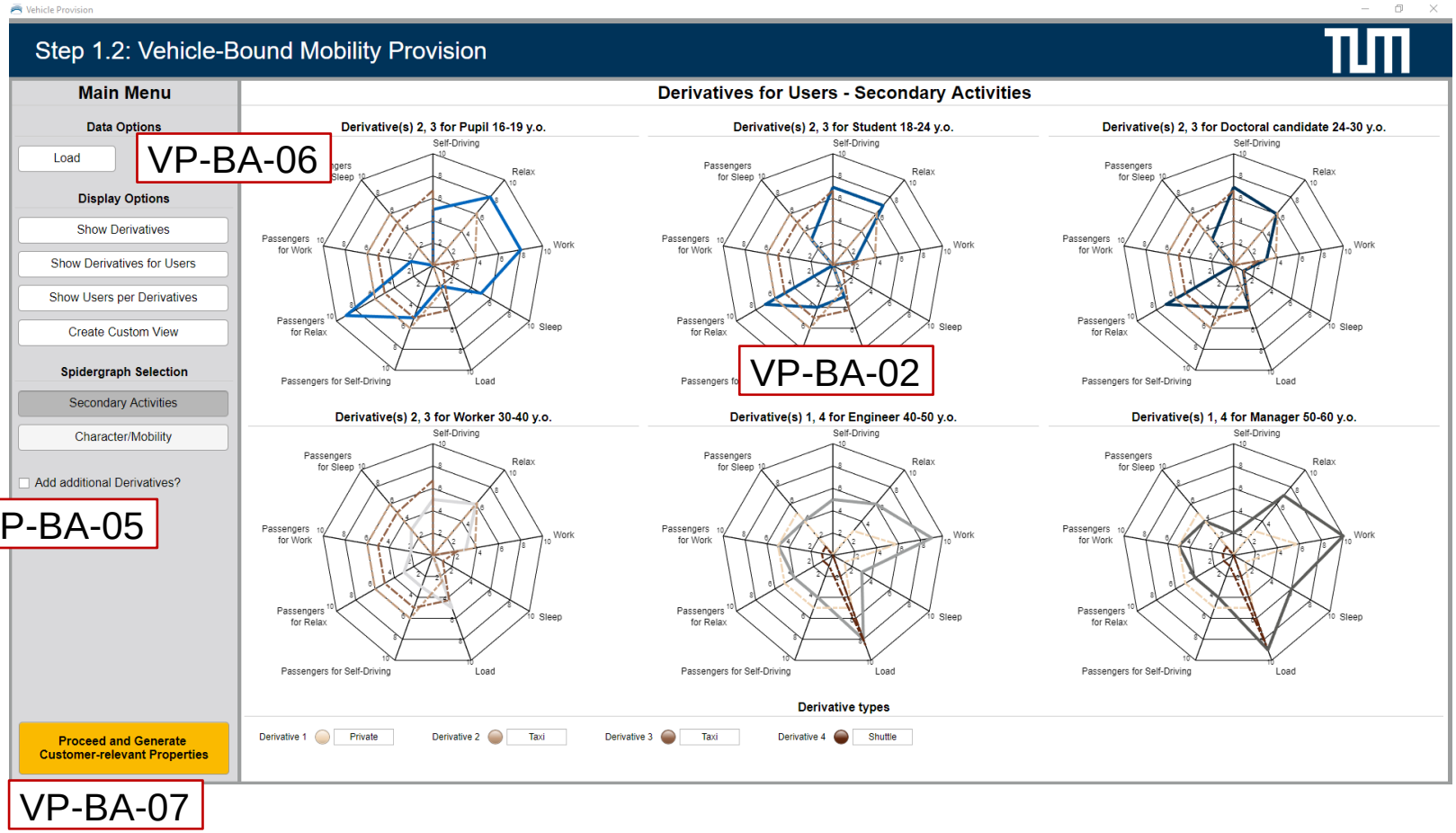
Step 1.2: Vehicle-Bound Mobility Provision

Anordnungen und Erfüllung Nutzerbedürfnisse:



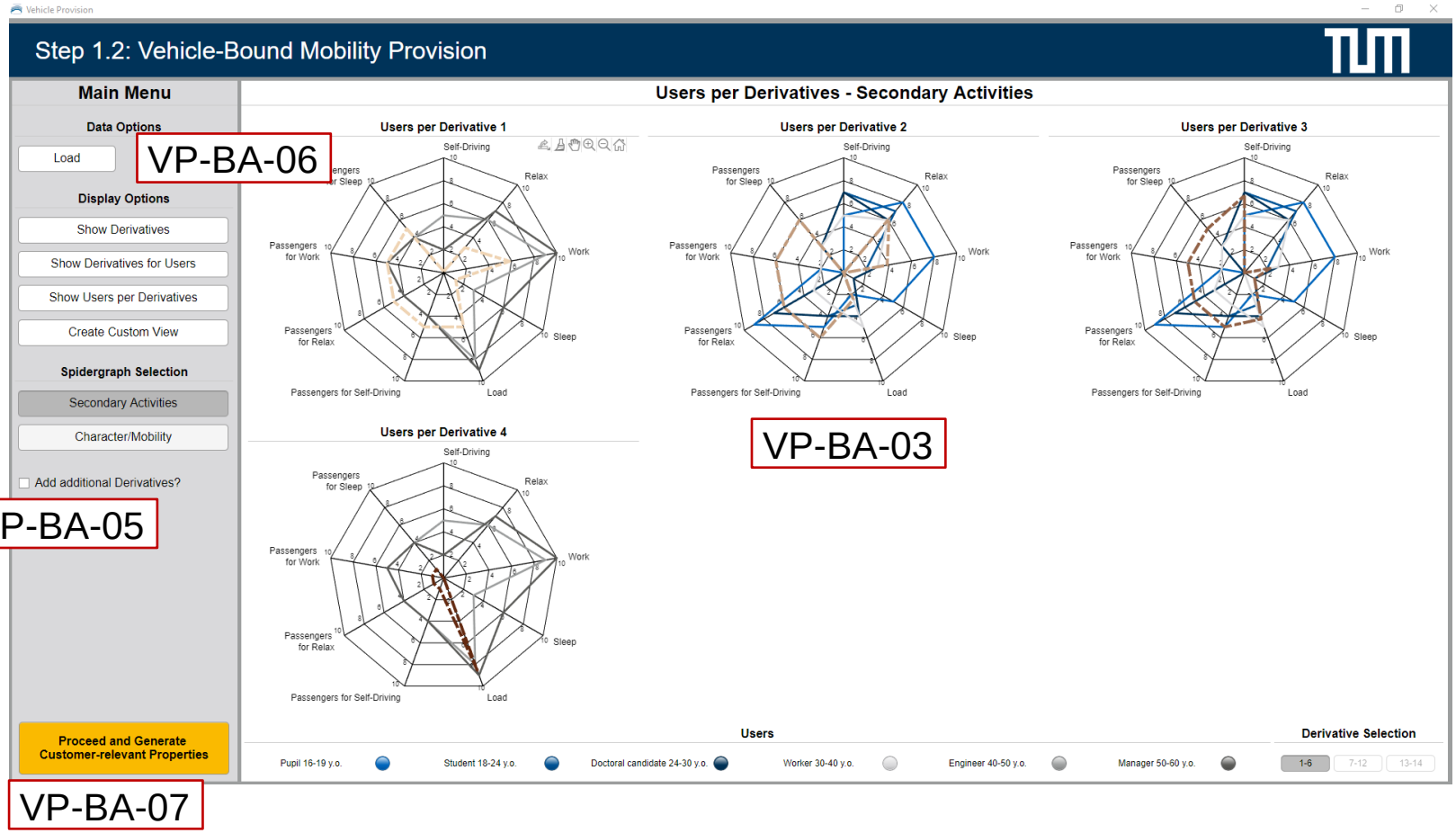
Step 1.2: Vehicle-Bound Mobility Provision

Anordnungen und Erfüllung Nutzerbedürfnisse:



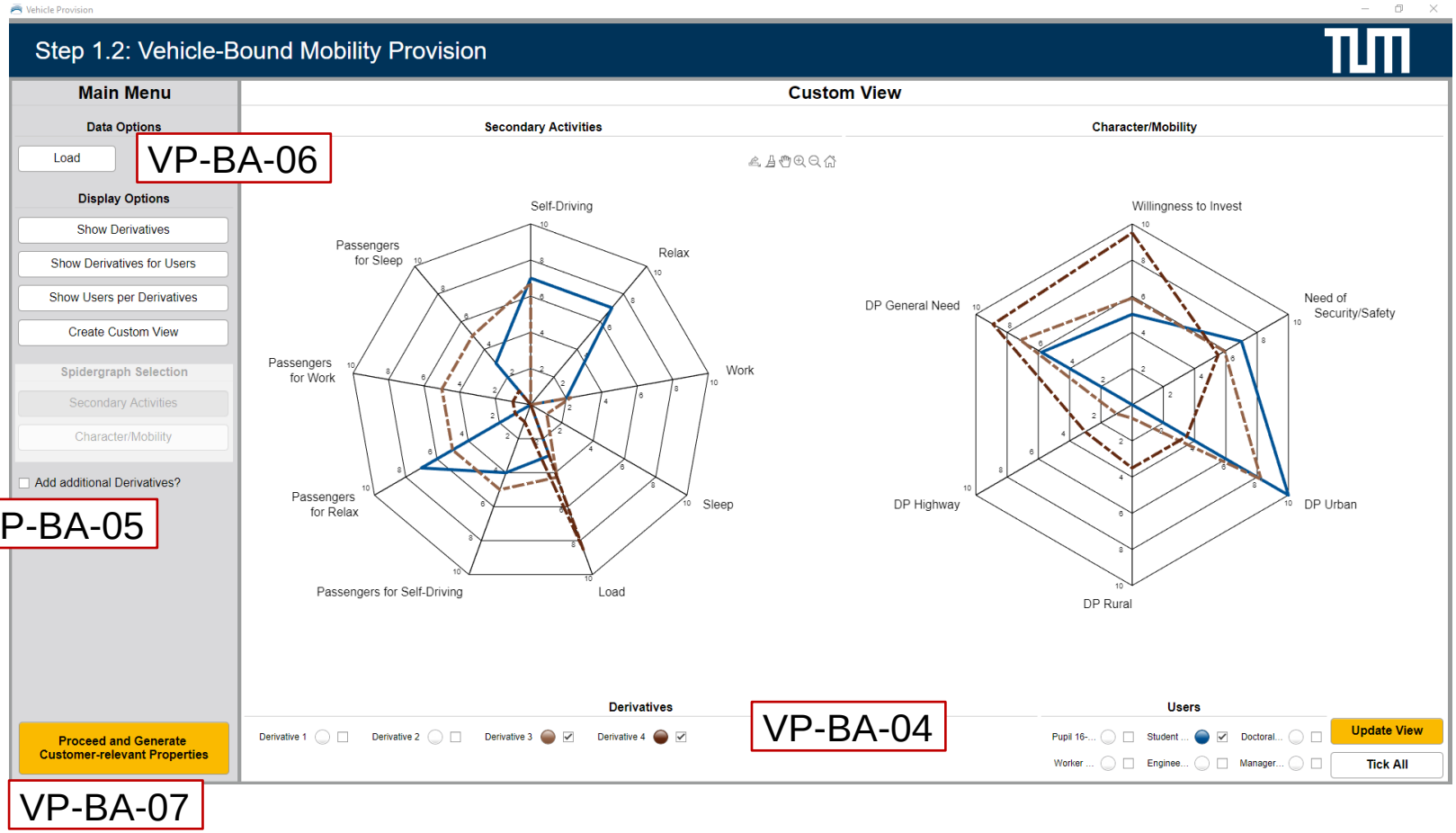
Step 1.2: Vehicle-Bound Mobility Provision

Anordnungen und Erfüllung Nutzerbedürfnisse:



Step 1.2: Vehicle-Bound Mobility Provision

Anordnungen und Erfüllung Nutzerbedürfnisse:



Step 2: Customer-Relevant Properties

Kontext:

- Dritte GUI, die zur Repräsentation des zweiten Berechnungsschritts verwendet wird
- Darstellung der Derivate und dazugehörigen Fahrzeugkonzepten
- Initiiert dritten Berechnungsschritt (Technische Eigenschaften der Fahrzeugkonzepte)

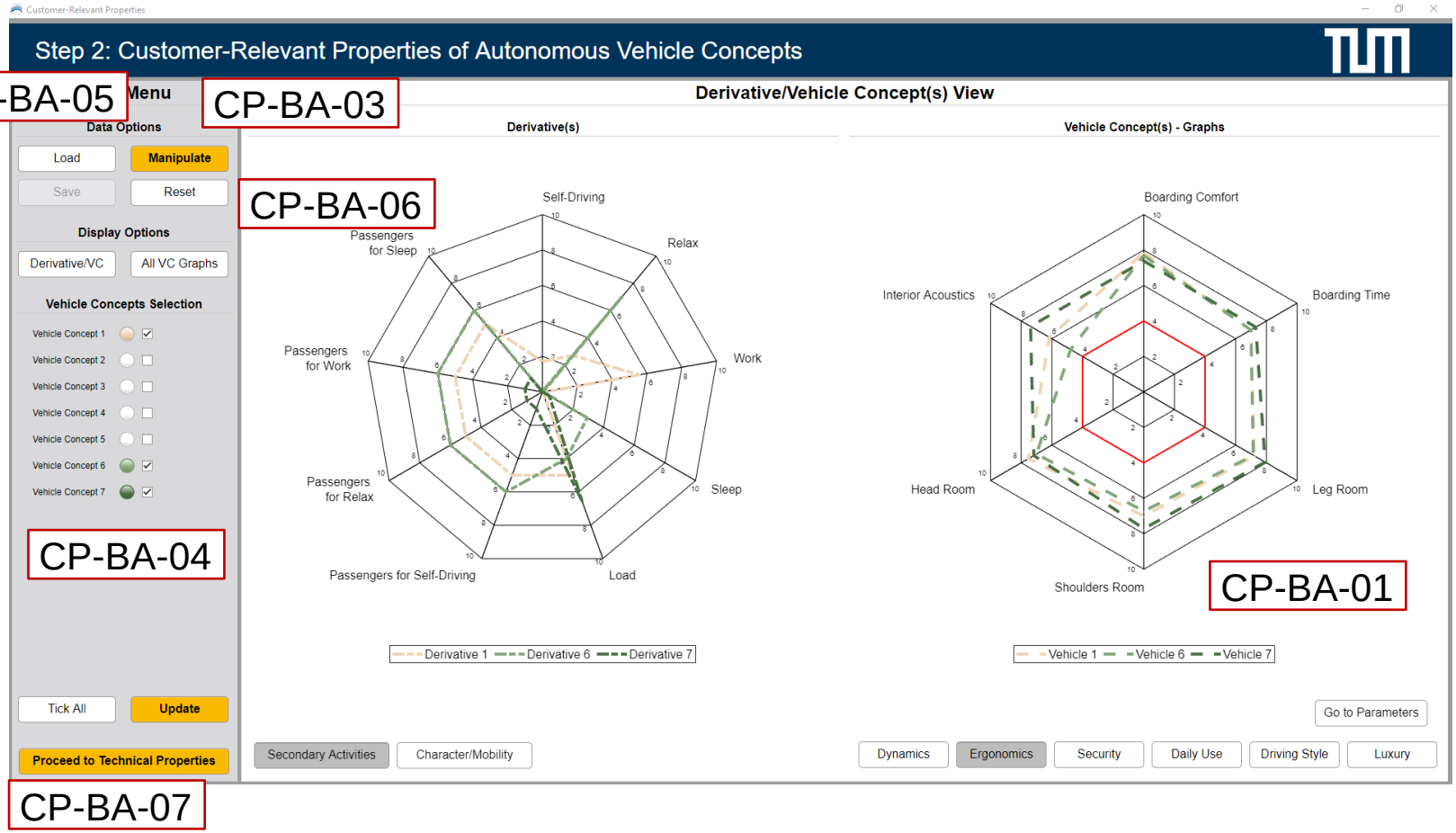
Entwicklungsaufgabe: (siehe Benutzeranforderungen)

Tabelle 4.7: Benutzeranforderungen Customer-Relevant Properties

GUI	Anforderung	Index
Customer-Relevant Properties	Anzeige des Fahrzeugkonzepts mit kundenwerten Eigenschaften	CP-BA-01
	Anzeige der Werte der kundenwerten Eigenschaften	CP-BA-02
	Manipulation der Werte	CP-BA-03
	Auswahl der Fahrzeugkonzepte	CP-BA-04
	Speicher- und Ladefunktion	CP-BA-05
	Zurücksetzen der Parameter	CP-BA-06
	Start der Berechnung der technischen Eigenschaften	CP-BA-07

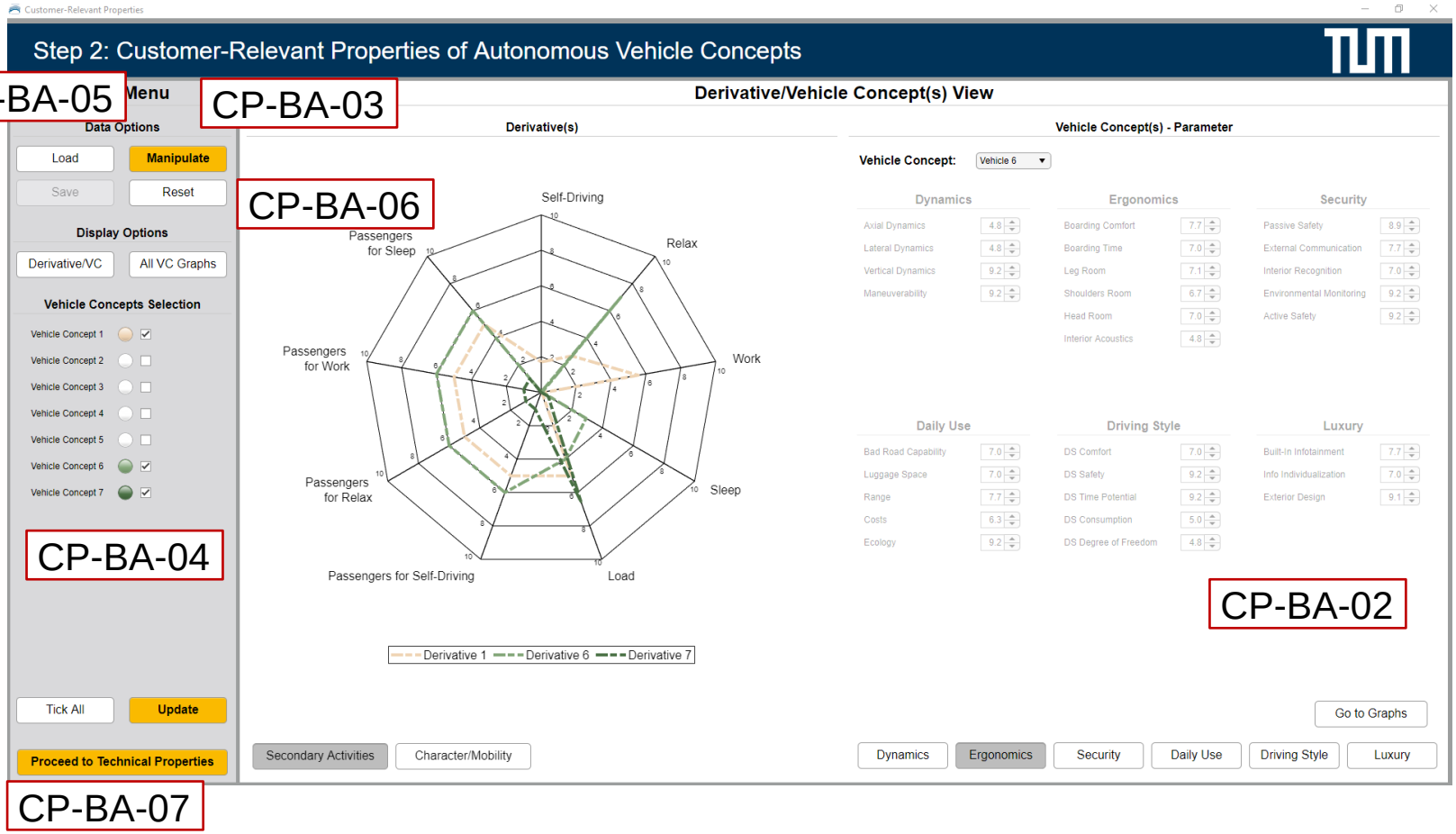
Step 2: Customer-Relevant Properties

Anordnungen und Erfüllung Nutzerbedürfnisse:



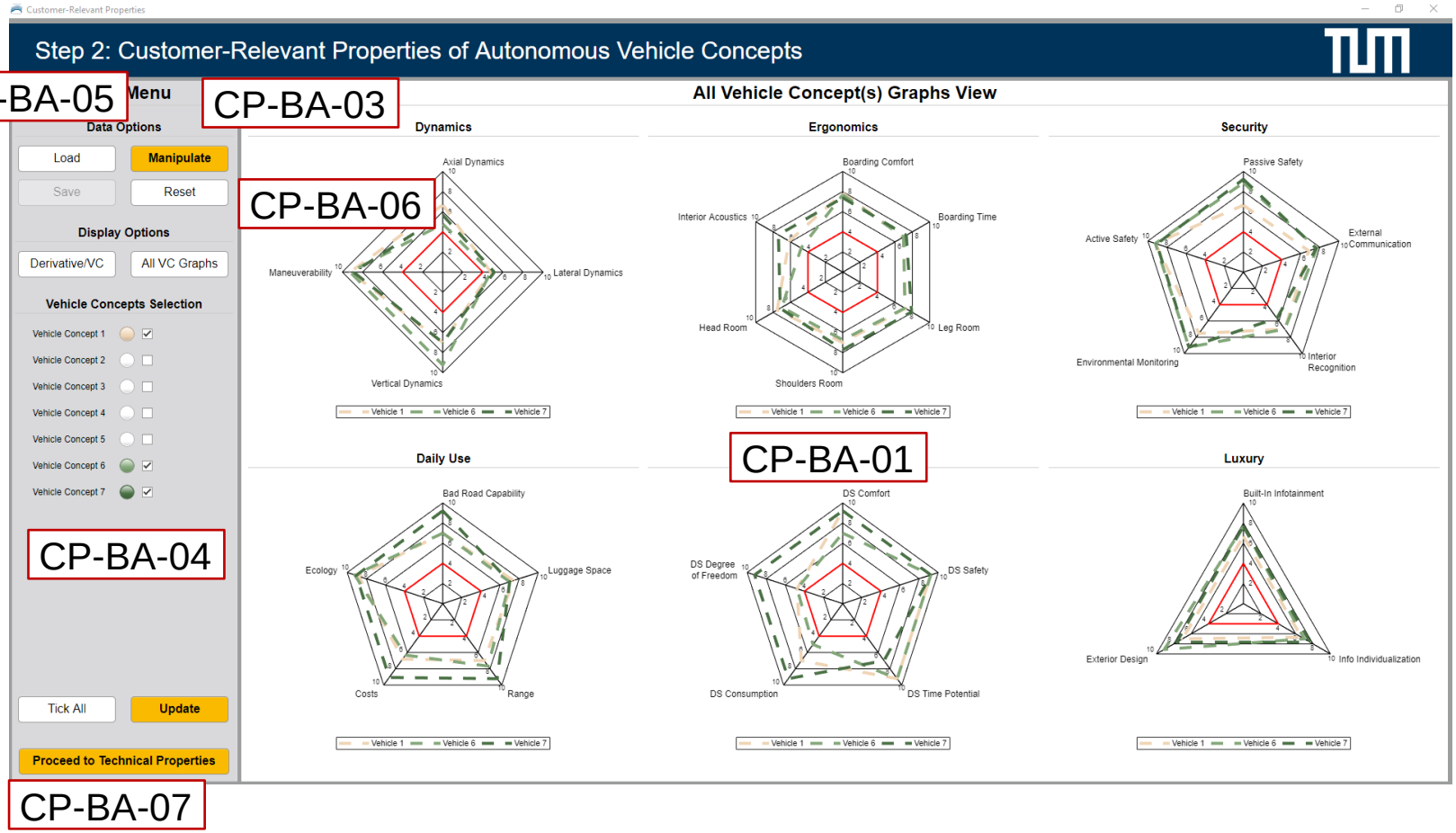
Step 2: Customer-Relevant Properties

Anordnungen und Erfüllung Nutzerbedürfnisse:



Step 2: Customer-Relevant Properties

Anordnungen und Erfüllung Nutzerbedürfnisse:

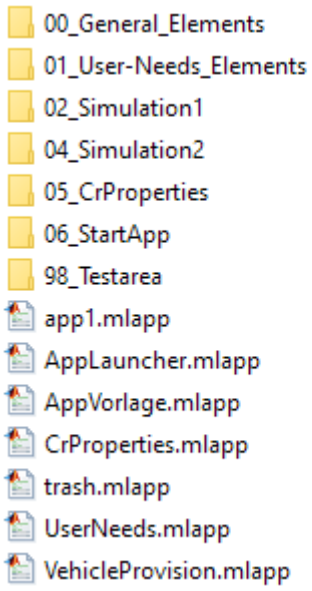


6. Zusätzliche Hilfe

- Ordnerstruktur
- Einbinden von Arbeitsordnern in Matlab

Ordnerstruktur

- Alle GUI auf einer Ebene in einem Ordner
- Diesen Ordner „99_GUI“ benennen und als Arbeitsverzeichnis in Matlab auswählen
- Jede GUI hat bei Bedarf seinen eigenen Ordner und Zugriff auf einen Simulationsordner
- Im Ordner „00_General_Elements“ werden allgemeine Dokumente abgelegt für jede GUI



Einbinden von Arbeitsordnern

- Ordner die vor dem automatischen Code-Aufbau aufgerufen werden sollen als private Variablen hinterlegen

```
properties (Access = private)
    oldpath = addpath('00_General_Elements\') % Needed for first build up of app (TUM image)
end
```

- In jeder GUI sind die zugehörigen Ordner in der StartUpFunction aufzurufen

```
            %Adding necessary folders to path
            addpath('00_General_Elements\')
%             addpath('01_User-Needs_Elements\') -> Add here the folder for
%             the specific app
```