

Abschlussprojekt

Jan.schulz@digitalcareerinstitute.org

Agenda

1. Features
2. Mockups
3. Minimum Viable Product (MVP)
4. Backend Features
5. Aufgabenteilung
6. Zeitplan
7. Digitalisierung Mockups
8. Erstellung Repository
9. Komponentendiagramm
10. Redux / Global State
11. Local Storage
12. Github Project-Board
13. Github-Zusammenarbeit
14. Coding

1. Features

- Identifiziere alle Features, die Deine App haben soll
- Schreibe alle Features ungeordnet in eine Liste

2. Mockups

1. Hol Dir ein paar Blätter Papier, ein paar Stifte
2. Erstelle Userinterface Graphiken (Mockups) von allen Features, die Du gefunden hast
3. Lege die Blätter Papier in der Reihenfolge, die Deinem Userflow entspricht

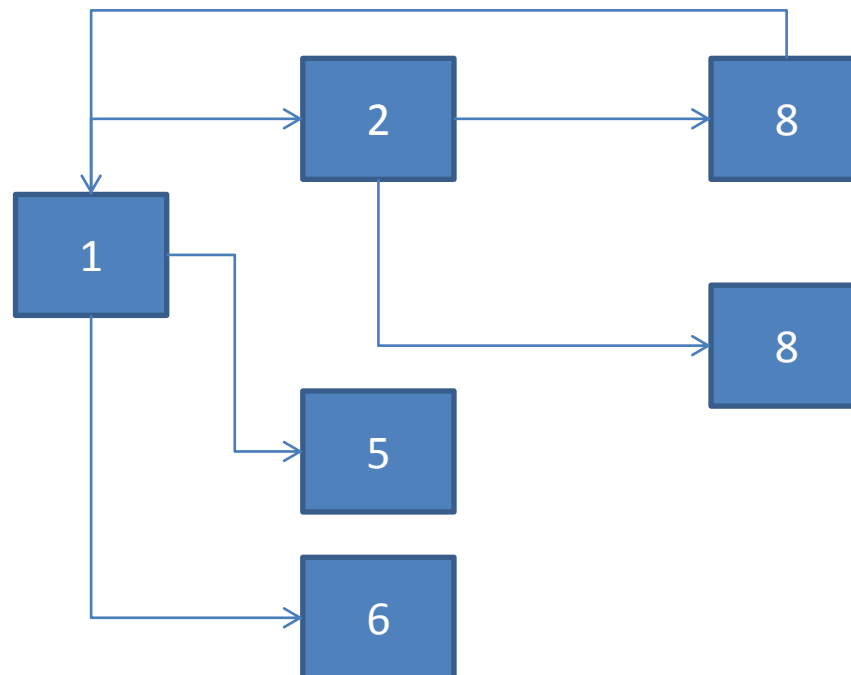
3. Minimum Viable Product (MVP)

1. Identifiziere die Features, welche Die Kernfunktionalität Deiner App darstellen
2. Eliminiere die anderen Features
3. Passe Deine Mockups an/ erstelle neue Mockups
4. Verändere ggfalls. Deinen Userflow

3. Minimum Viable Product (MVP)

1. Gib jedem Mockup eine Id. Erstelle ein Userflow-Diagramm, basierend auf Deinen Ids.

Bsp:



4. Backend Features

1. Features soll Dein Backend beherrschen? Gib jedem Feature eine eigene Resource und schreibe diese auf. Bsp:
„Login“ -> /login
„Registrieren“ -> /signup
„Upload“ -> /upload
2. Für jedes Feature, erstelle einen konzeptionellen Beispiel-Request. Zum Beispiel:

```
POST Login -> /login
{ user: "jan", password: "foobar" }
RESPONSE
{ error: 0 }
```

4. Backend Features

Beispiel 1

Req:

POST /user

body: { username: "jan", password: "foobar" }

Res (Success):

{ error: 0 }

Res (Error):

{ error: 1001, message: "username already exists" }

4. Backend Features

Beispiel 2

Req:

DELETE /user/:id

Res (Success):

{ error: 0 }

Res (Error):

{ error: 1002, message: "user deleted" }

4. Backend Features

Beispiel 3

Req:

POST /fileupload
{ username: "jan" }
+ Data-Stream

Res (Success):

{ error: 0 }

Res (Error):

{ error: 1002, message: "fileupload done" }

5. Aufgabenteilung

1. Identifiziere mögliche Aufgabengebiete.
 - Frontend / Backend
 - Design
 - Testing
 - Texte
 - Recherche
 - ...
2. Sprich mit Deinen Teamkollegen, wer möchte was machen?
3. Entscheide, wer was macht

6. Zeitplan

1. Du hast 6-8 Wochen Zeit, um **alles** fertig zu bekommen.
2. Überlege Dir 3-5 Milestones, Die in dieser Zeit erreicht werden sollen.
3. Ordne jedem Milestone ein Datum zu.

7. Digitalisierung Mock-Ups

1. Suche Dir ein Tool, in welchem Du Deine Konzept digitalisieren kannst
 - Libre Impress
 - MS Powerpoint
 - Balsamiq
 - ...
2. Digitalisiere Deine Mockups und speichere die Tool-Datei sowie ein PDF

8. Erstellung Repository

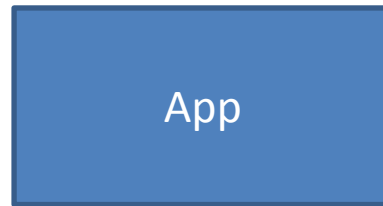
1. Erstelle ein Github-Repository und lade Deine Teamkollegen ein
2. Erstelle zwei Ordner: „code“ und „concept“
3. Pushe alle Deine Konzept-Dateien in das Concept Verzeichnis.
4. In Deinem „code“ Verzeichnis, erstelle zwei Unterordner „app“ und „experiments“.
5. In Deinem „app“-Verzeichnis, erstelle eine leere HalloWorld-App.
6. In Deinem experiments, pushe alle Prototypen bzw. bisherigen Code-Beispiele bzgl. Deines Projektes.

9. Komponentendiagramm

1. Überlege Dir, welcher Mockup auf welcher Frontend-Route zu finden sein soll. Bsp:
 - Deine Startseite ist auf „/“ zu finden
 - Deine Profilseite ist auf „/profile“ zu findenusw.
2. Anhand Deiner Mockups, überlege Dir, welche Top-Komponenten (also Komponente, die auf höchster Ebene unter Deinen App-Komponenten zu finden sein sollen.

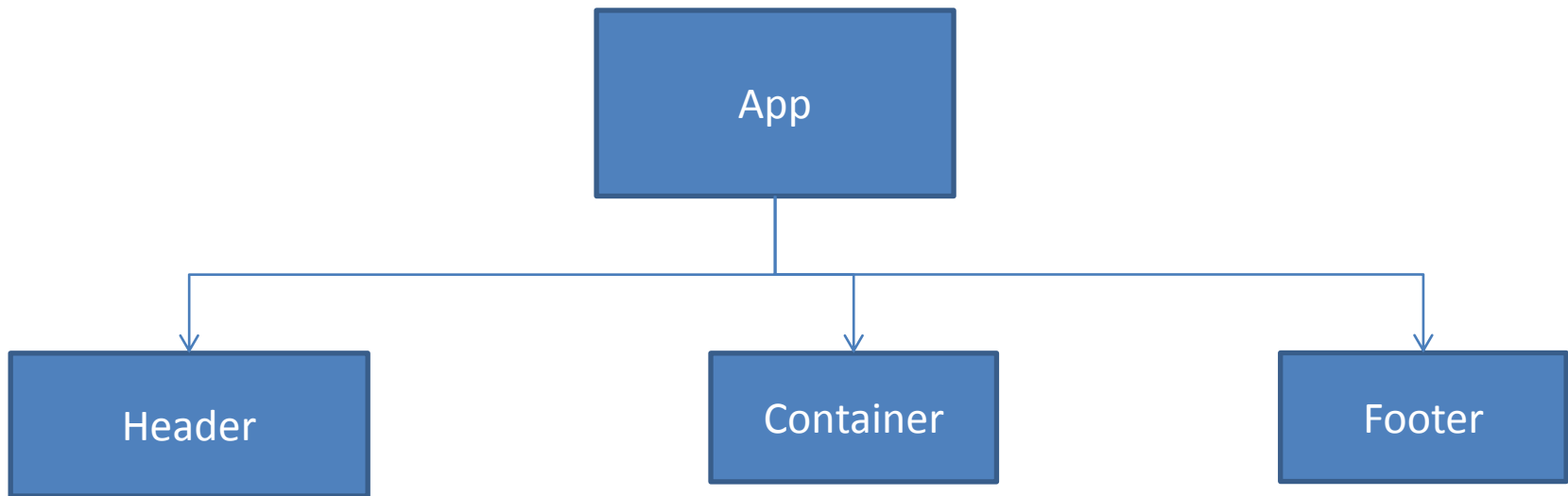
9. Komponentendiagramm

2. Bsp: Der Root-Komponent



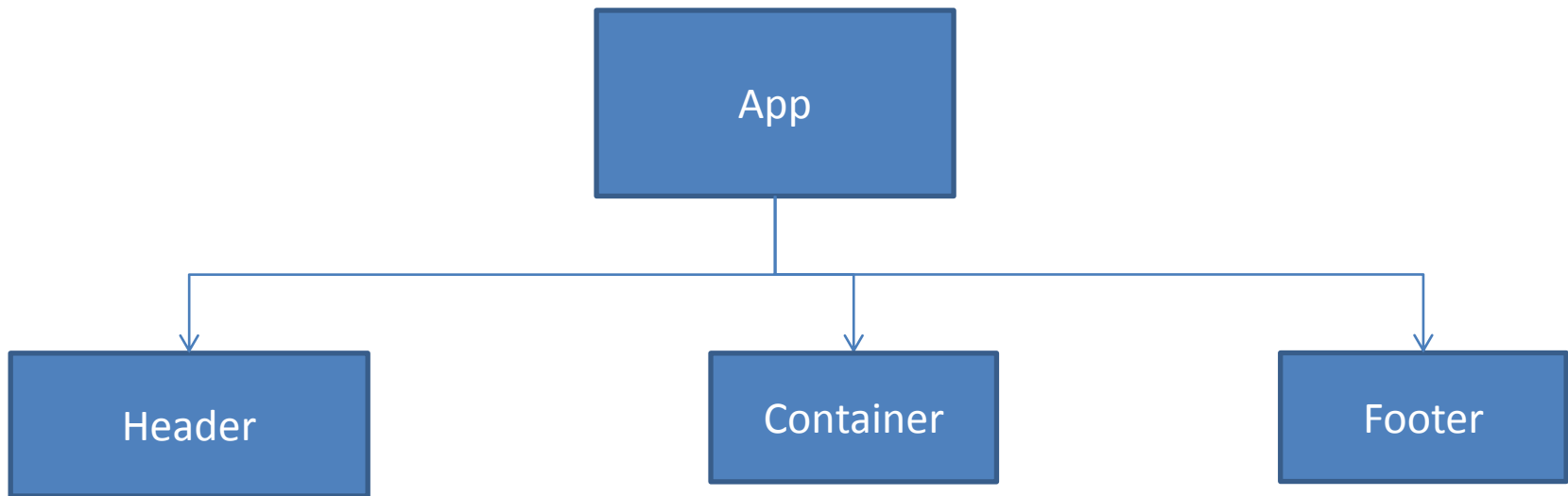
9. Komponentendiagramm

2. Bsp: Header, Container und Footer



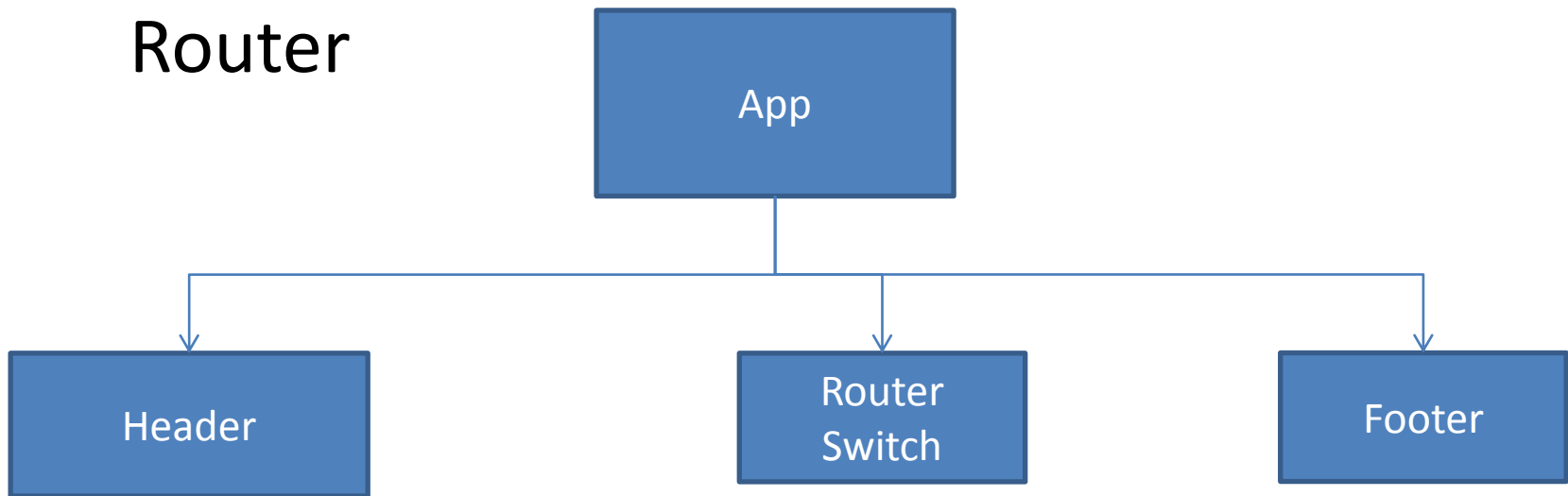
9. Komponentendiagramm

2. Bsp: Weil wir verschiedene Routes brauchen



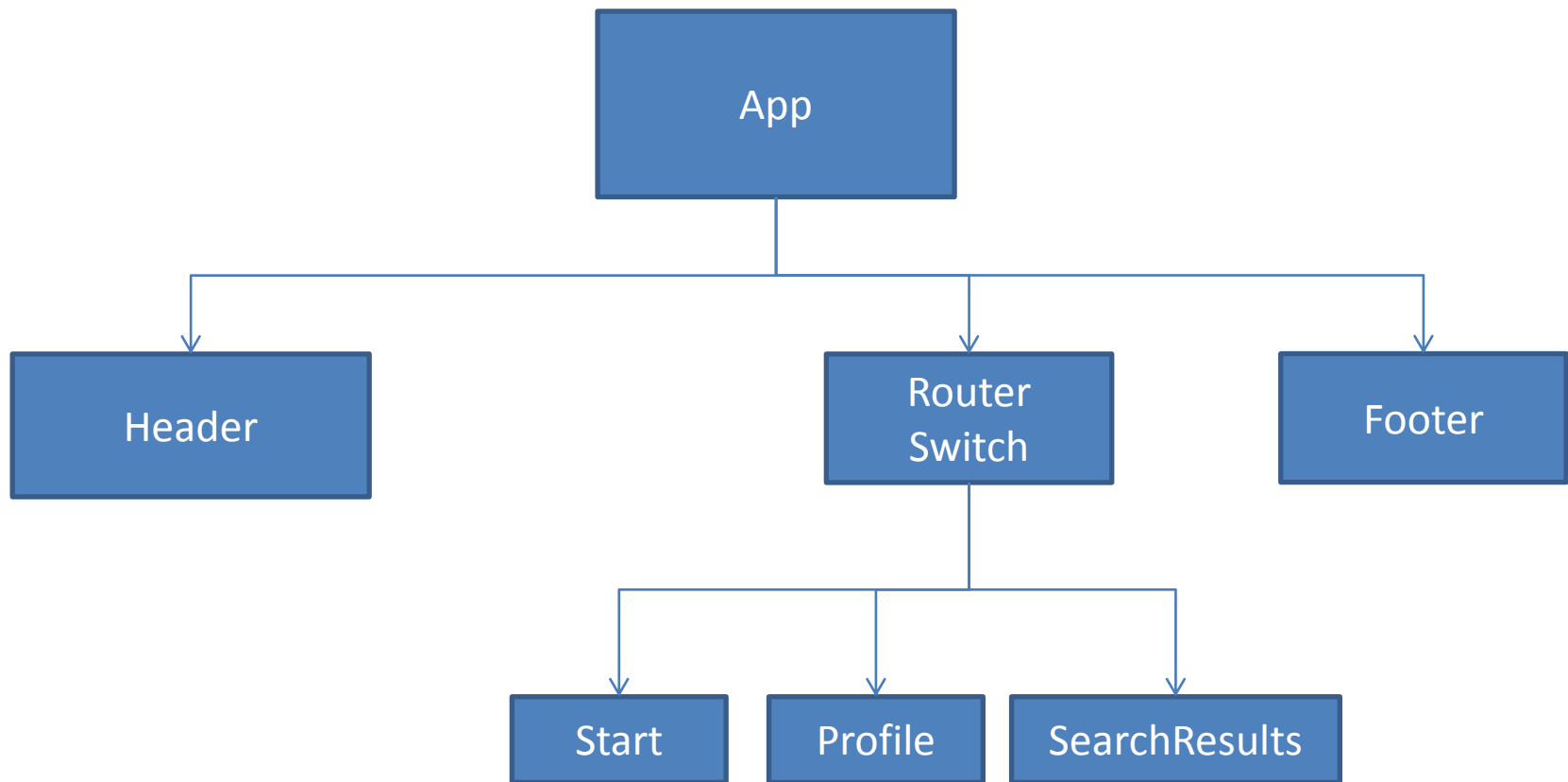
9. Komponentendiagramm

2. Bsp: ... Bauen wir über den Container einen Router



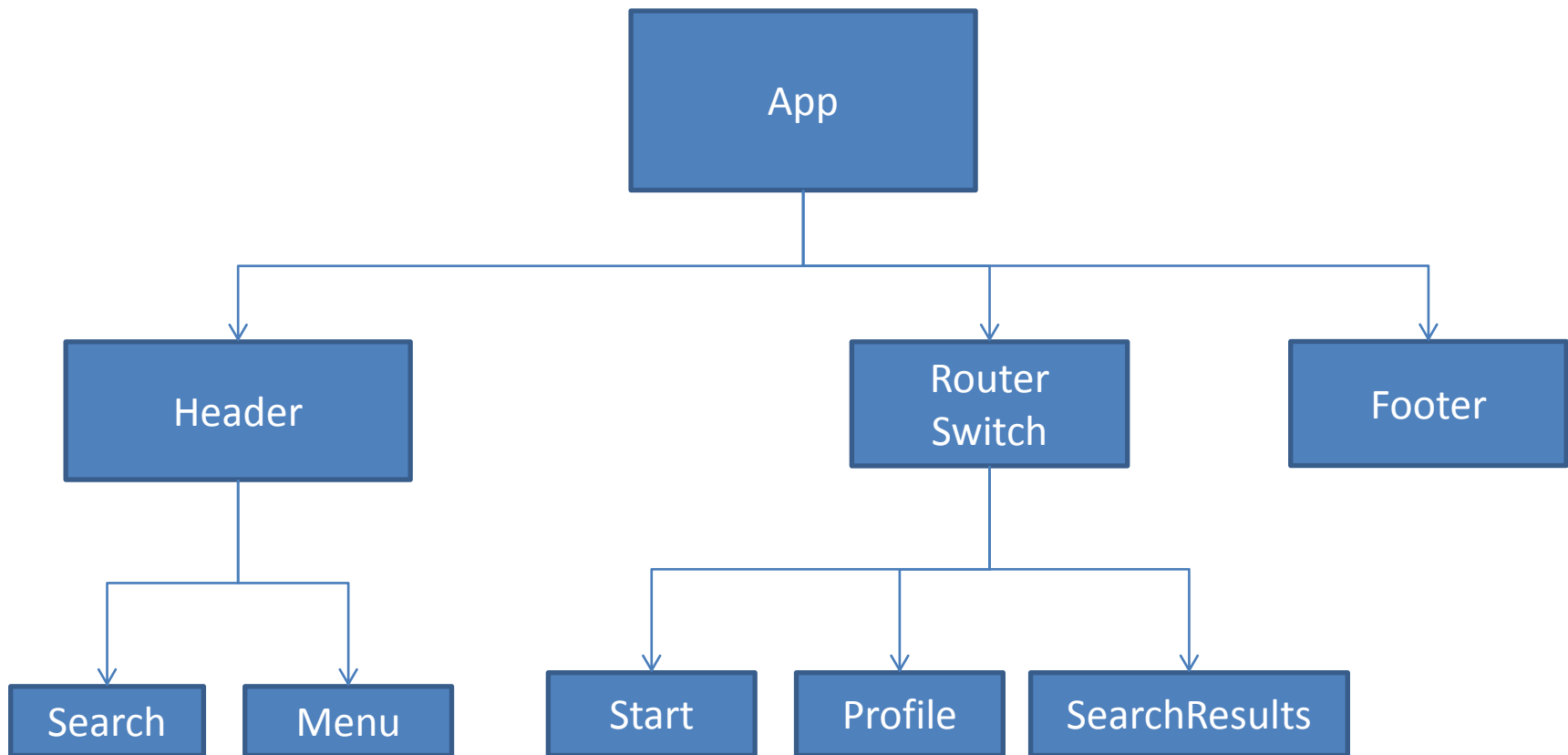
9. Komponentendiagramm

2. Bsp: und für die Start-, Profil- und SearchResultseite



9. Komponentendiagramm

2. Bsp: Der Header wird unterteilt in Search und Menu

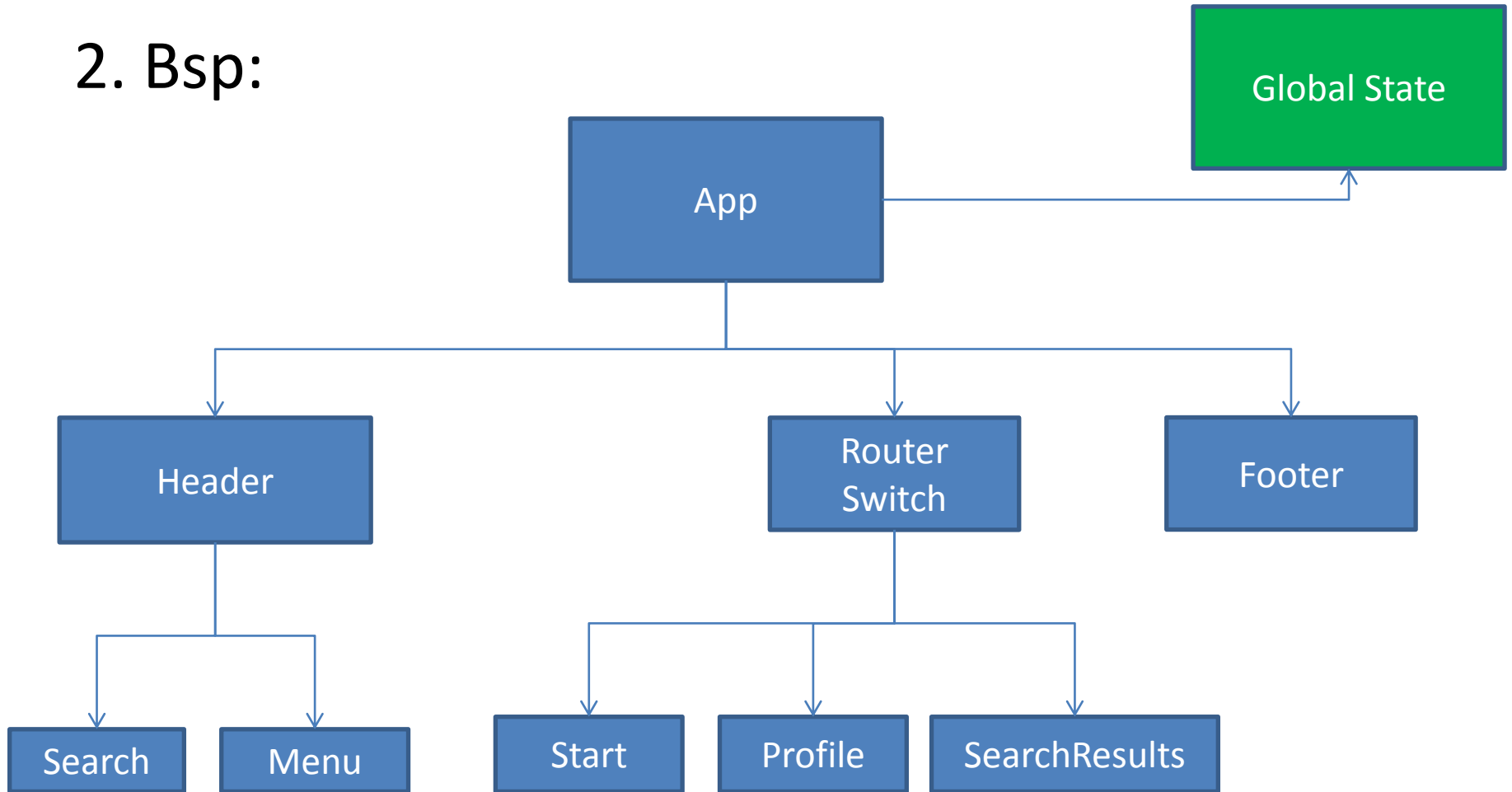


10. Redux / Global State

1. Was soll im global State gespeichert werden?
2. Welche Komponente soll sich neu-rendern, wenn welche globale Variable geändert wird?

10. Redux / Global State

2. Bsp:



11. localStorage

- Was soll im localStorage gespeichert werden?
Bsp: ist man eingeloggt?

12. GitHub – Project Board

1. In Deinem Github-Repo, gehe auf „Projects“ und erstelle ein neues GitHub-Project mit dem Namen „DCI-Finalproject“.
2. Wähle Kanban-Basic aus.
3. Klicke nun auf Issues und erstelle einen ersten Issue „Überschrift erstellen“. Bevor Du auf „Submit New Issue“ klickst, wähle einen Assignee aus und das Project „DCI-Finalproject“.
4. Klicke Dein Issue nochmal an und rechts bei Projects, weise den Issue der Todo-Sparte des Kanban zu.
5. Gehe zu „Projects“ und klicke dort auf „DCI-Finalproject“.
6. Schaue Dir Dein Kanban-Plan an.

13. GitHub – Zusammen arbeiten

- **Teamleiter** coden auf Ihrem eigenen Branch und auf dem Master-Branch
- **Teammitglieder** coden auf nur auf ihrem eigenen Branch

13. GitHub – Zusammen arbeiten

Übungen

Bsp 1: Teamleiter

1. Erstelle ein neues Kanban-Issue – „Hallo World“.
2. Setze den Issue auf „In Progress“.
3. Clone/Pulle Dir das Repository.
4. Gehe auf den Master-Branch.
5. Ändere dort eine Datei – schreibe „Hallo World“ hinein.
6. Commit und Pushe.
7. Setze Den Issue auf „Done“.

13. GitHub – Zusammen arbeiten

Übungen

Bsp 2: Teamleiter

1. Erstelle ein neues Kanban-Issue – „Roter Header“.
2. Setze den Issue auf „In Progress“.
3. Clone/Pulle Dir das Repository.
4. Erstelle ein neuen Feature-Branch “roter-header”.
5. Ändere dort eine Datei.
6. Committe.
7. Gehe auf den Master-Branch.
8. Merge „Roter Header“ in den Master-Branch.
9. Setze den Issue auf „Done“.

13. GitHub – Zusammen arbeiten

Übungen

- Folgende Übungen bitte mit allen 2-3 Teammitgliedern machen
- Jeweils mit „Impressum“, „Hallo World“, „Hallo Sonne“.
- Es geht in jedem Szenario darum, eine Datei zu ändern.

13. GitHub – Zusammen arbeiten

Bsp 3 (Teil 1/3): Teamleiter

1. Erstelle ein neues Kanban-Issue „Impressum“.
2. Assigne ein Teammitglied und setze den Status auf Todo.

13. GitHub – Zusammen arbeiten

Bsp 3 (Teil 2/3): Teammitglied

1. Lies Deine Email / Schau auf Github unter Issues oder Projects.
2. Pülle/Clone das Repository.
3. Setz den Status des Issue auf „In-Progress“.
4. Erstelle einen neuen Branch „Impressum“.
5. Ändere eine Datei, schreibe „Impressum“ hinein.
6. Commit und Pushe deinen Branch.
7. Setze den Status von Deinem Issue auf „Done“.

13. GitHub – Zusammen arbeiten

Bsp 3 (Teil 3/3): Teamleiter

1. Lies Deine Email / Schau auf Github unter Issues oder Projects. Schaue danach auf „Commits“.
2. Pulle/Clone das Repository.
3. Gehe auf den Master-Branch.
4. Merge den „Impressum“-Branch in den Master-Branch.
5. Commit und Pushe.

14. Coding

1. Auf Basis Deines Konzepts (Mockups + Komponentendiagramm), baue Deine Komponenten. Schreibe überall nur ein div bzw. einen ersten String rein z.B. “Das ist der Header“, „Das ist das Menu“, etc.
2. Implementiere den Frontend-Router so, dass Du durch Deine Seite hindurch navigieren kannst.
3. Für das Backend: implementiere Deine geplanten Routen und teste sie Postman.
4. Implementiere den Rest.

HAVE FUN CODING / HAPPY HACKING