

ASDFQ

Technical requirements

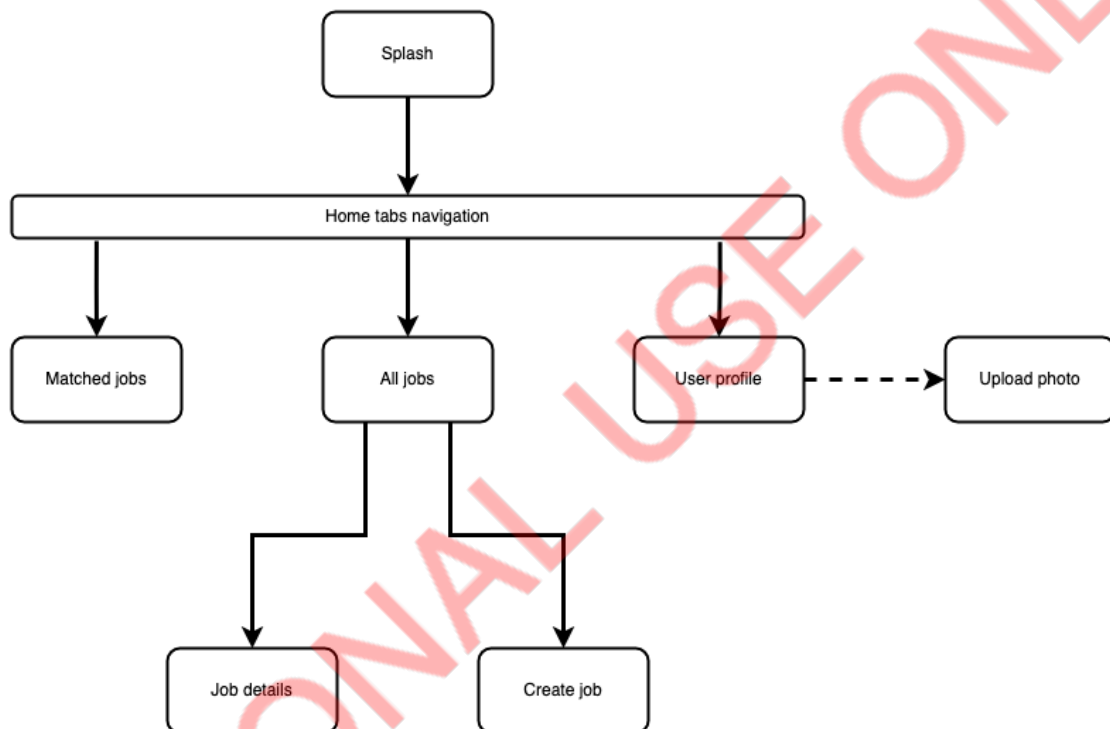
PERSONAL USE ONLY

CONTENT

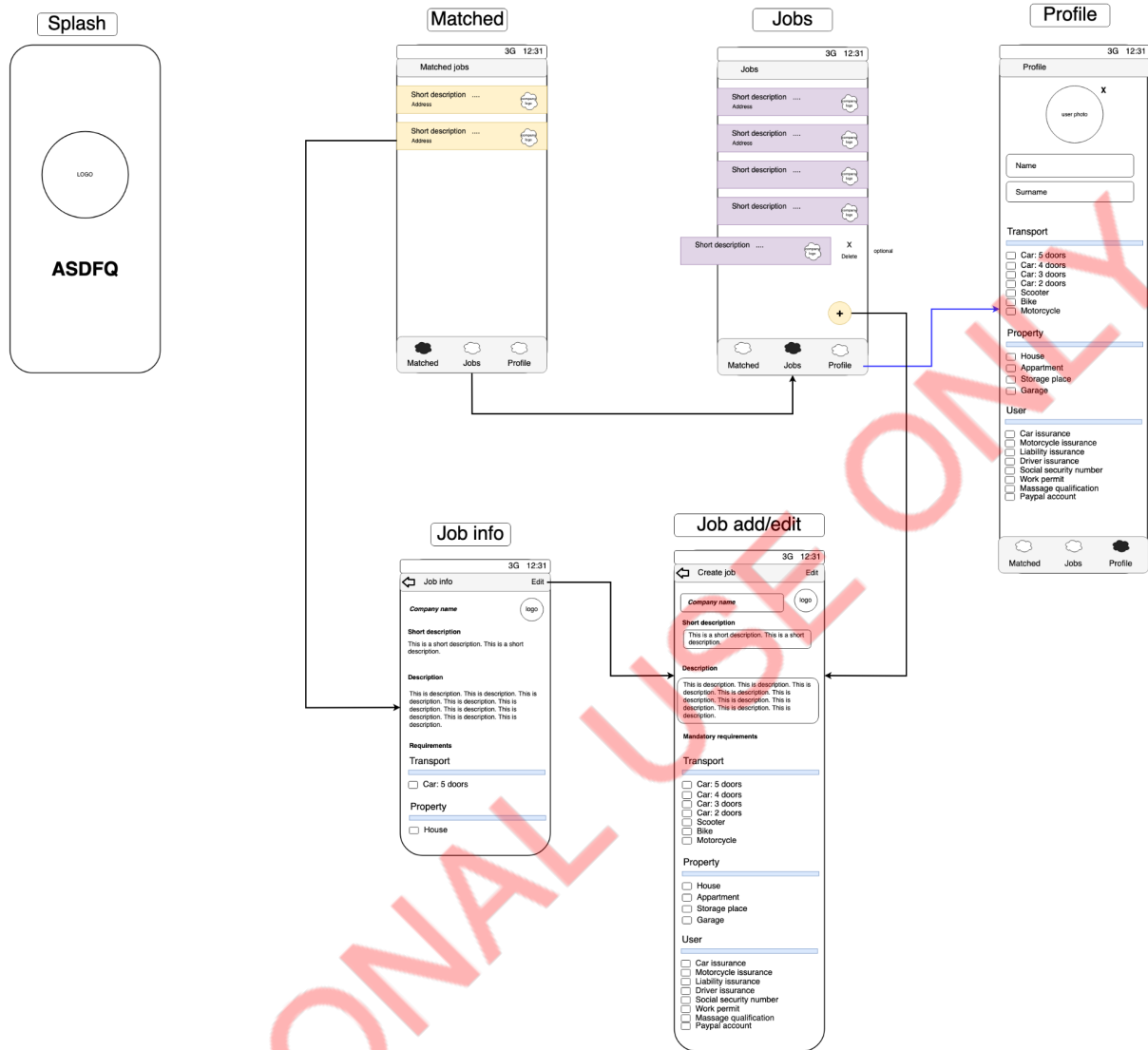
CONTENT	1
SCREENS, NAVIGATION	2
DATA MODEL	4
UserProfile	4
Job	4
SKILLS	5
enum Car {	5
enum User {	5
enum Property {	5
MATCHING ALGORITHM	6
BEFORE GETTING STARTED	7
USE CASES	8
Happy flow: match user and job	8
Unhappy flow: no match for user and job	8
TECHNICAL SOLUTION	9
RELEASE CHECKLIST	10

SCREENS, NAVIGATION

1. Splash screen
2. Matched screen
3. Job view screen
4. Job create/edit screen
5. User profile screen



Pic. Desirable navigation in the app



Pic. Screens mockup with navigation elements

DATA MODEL

UserProfile

- name
- surname
- skills
- *photo*

Job

- companyName
- shortDescription
- longDescription
- mandatorySkills
- *companyLogo*

PERSONAL USE ONLY

SKILLS

enum Car {

```
FIVE_DOOR_CAR = 'FIVE_DOOR_CAR';
FOUR_DOOR_CAR = 'FOUR_DOOR_CAR';
THREE_DOOR_CAR = 'THREE_DOOR_CAR';
TWO_DOOR_CAR = 'TWO_DOOR_CAR';
SCOOTER = 'SCOOTER';
BIKE = 'BIKE';
MOTORCYCLE = 'MOTORCYCLE';
```

}

enum User {

```
CAR_ISSURANCE = 'CAR_ISSURANCE';
MOTORCYCLE_ISSURANCE = 'MOTORCYCLE_ISSURANCE';
LIABILITY_ISSURANCE = 'LIABILITY_ISSURANCE';

DRIVER_LICENSE = 'DRIVER_LICENSE';

SOCIAL_SECURITY_NUMBER = 'SOCIAL_SECURITY_NUMBER';
WORK_PERMIT = 'WORK_PERMIT';

MESSAGE_QUALIFICATION = 'MESSAGE_QUALIFICATION';

PAYPAL_ACCOUNT = 'PAYPAL_ACCOUNT';
```

}

enum Property {

```
HOUSE = 'HOUSE';
APPARTMENT = 'APPARTMENT';
STORAGE_PLACE = 'STORAGE_PLACE';
GARAGE = 'GARAGE';
```

}

MATCHING ALGORITHM

Super straightforward matching.

Considering both skill set are array entity:

- loop through all the jobs
- loop through mandatorySkills in order to transform mandatorySkills to boolean type
- loop through mandatorySkills boolean representation

Algo can be improved in many ways, but it was not required in the original task.

```
const profile = { skills: [] };
const matchedJobs = jobs.filter((job: { mandatorySkills: [] }) => {
  const matched = mandatorySkills
    .map((mandatorySkill) => profile.skills.find((userSkill) => userSkill === mandatorySkill))
    .filter((possiblyMatchedSkill) => possiblyMatchedSkill);

  return matched;
});
```

BEFORE GETTING STARTED

Screens navigation

Logo

Icons

- Matched jobs
- All jobs
- Profile

Colors

- Primary color:
- Secondary color:

PERSONAL USE ONLY

USE CASES

Happy flow: match user and job

a.

Unhappy flow: no match for user and job

PERSONAL USE ONLY

TECHNICAL SOLUTION

1. ReactNative 0.64 or higher, started with expo. Ejected
2. Navigation: react-navigation
3. Libraries:
 - a. Splash-boot-screen
 - b. Swipeable list item (optional)
 - c. Material UI or similar
 - d. String-based icon generator
4. Lint:
 - a. Eslint + prettier
 - b. Babel-module-resolver (optional)
5. Data management:
 - a. No special libraries
 - b. Providers + hooks
 - c. AsyncStorage if necessary
6. No networking

RELEASE CHECKLIST

1. List of possible use cases is ready
2. All screens is ready, navigation works as expected
3. Unit test for matching algo
4. Make sure Android app ready to go
5. README if necessary
6. Github contains documentation
7. Video proof for both iOS and Android
8. E2E testing (optional)

PERSONAL USE ONLY