TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Circles

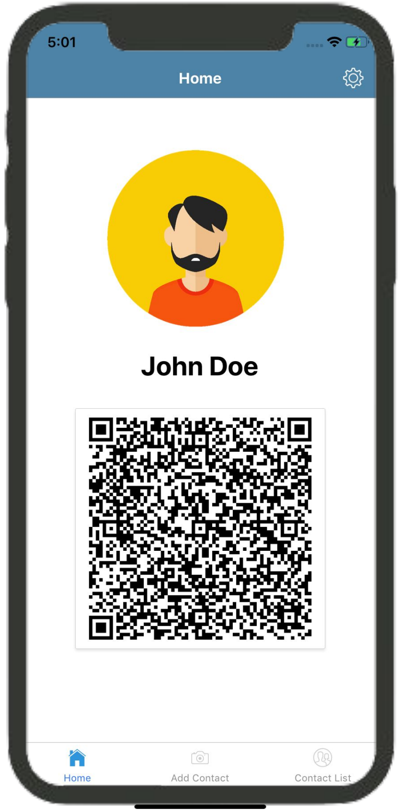
Application Development for Mobile and Ubiquitous Computing

Dresden, February 1st 2019

Adriano Batista
Maria Beatriz Moreira

Idea

- Add contact information using QR code
- Personalize what you want to share (eg. phone number, address, facebook page…)

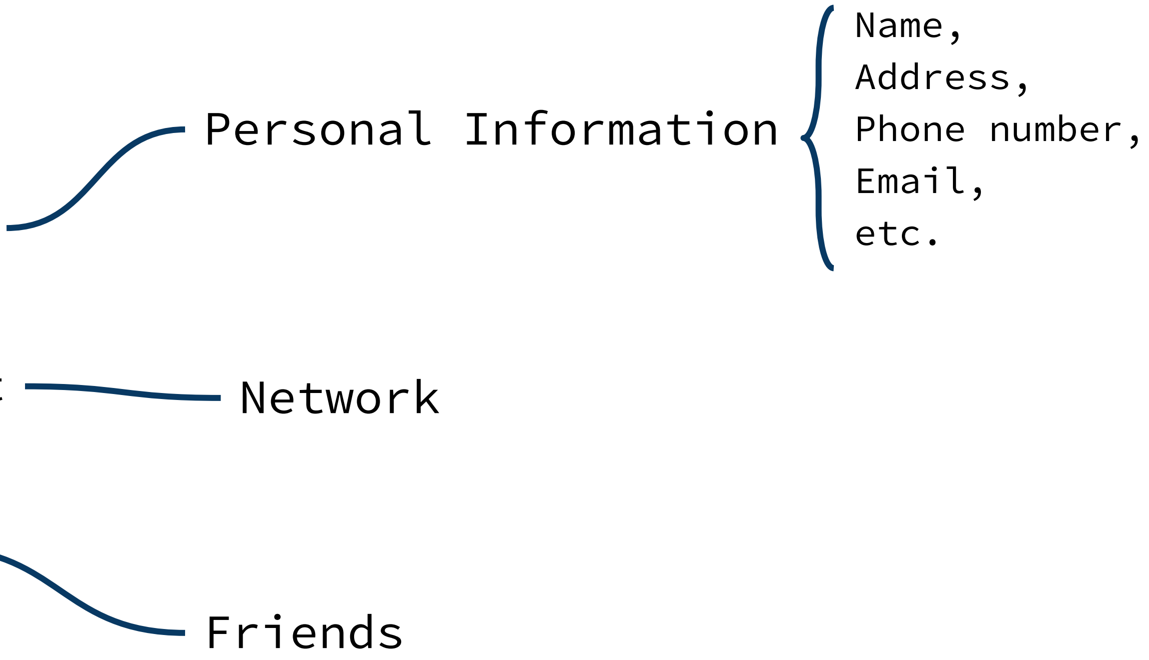


Contexts

• Personal Context	Personal Information
• Technical Context	Network
• Social Context	Friends

Contexts

- Personal Context
- Technical Context
- Social Context



Contexts

- Personal Context
- Technical Context
- Social Context

Personal Information

Network

Network state (Wifi, cellular, 3g, 4g) is captured by NetInfo package

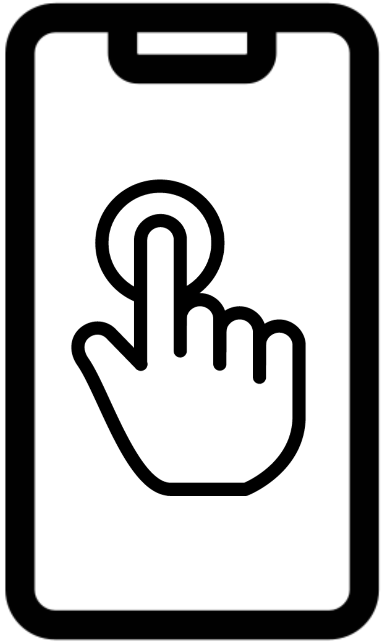
Friends

Contexts

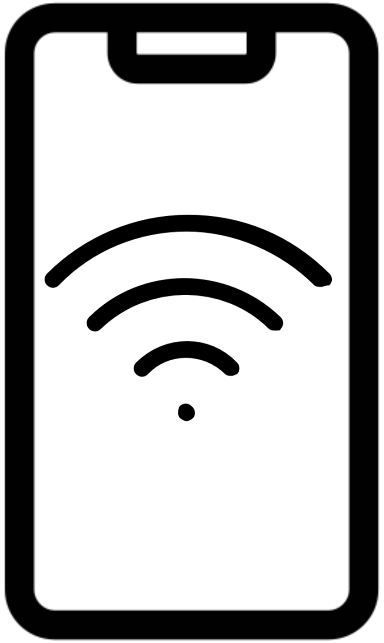
- Personal Context — Personal Information
- Technical Context — Network
- Social Context — Friends { User’s previously scanned contacts

Challenges

Usability

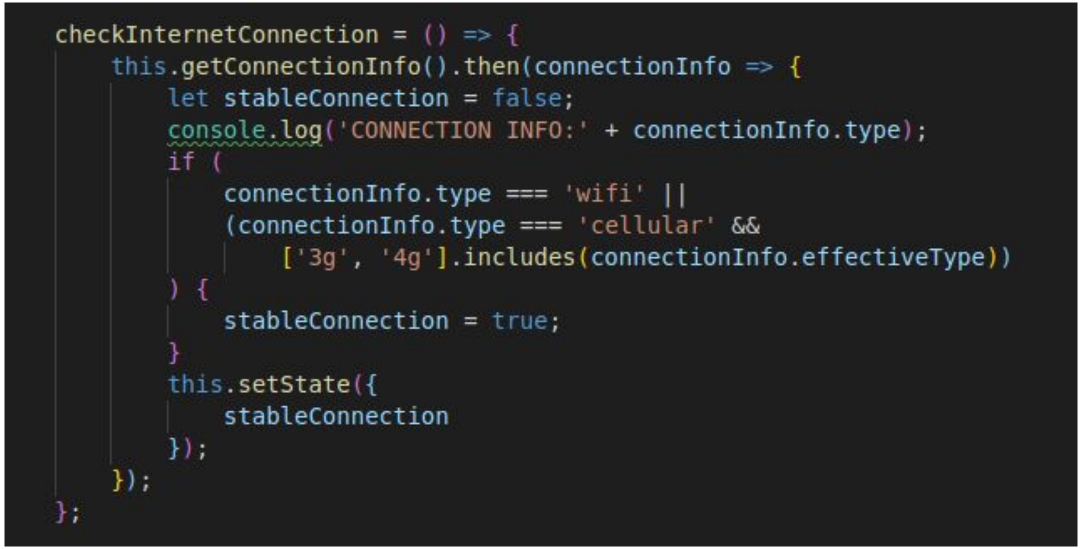


Connectivity



Implemented Challenges - Connectivity

- Use the NetInfo package to detect network state: Wifi, cellular, 2g, 3g, 4g or none.

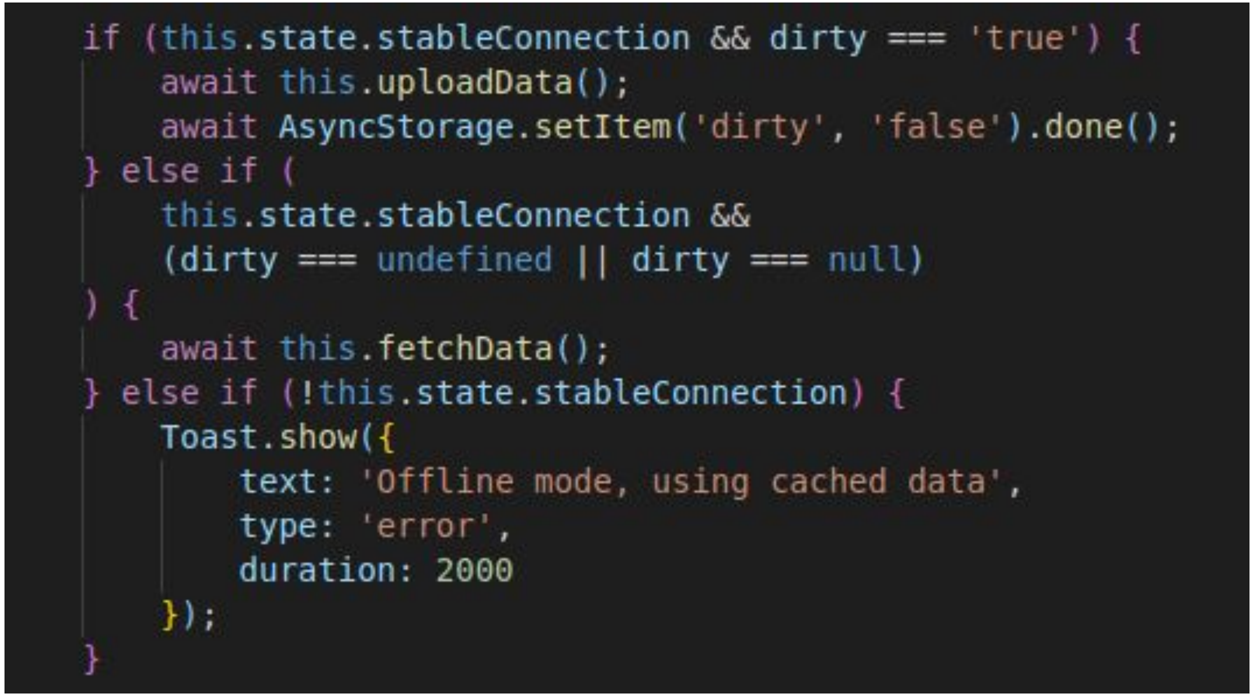


```
checkInternetConnection = () => {
    this.getConnectionInfo().then(connectionInfo => {
        let stableConnection = false;
        console.log('CONNECTION INFO:' + connectionInfo.type);
        if (
            connectionInfo.type === 'wifi' ||
            (connectionInfo.type === 'cellular' &&
                ['3g', '4g'].includes(connectionInfo.effectiveType))
        ) {
            stableConnection = true;
        }
        this.setState({
            stableConnection
        });
    });
};
```

Implemented Challenges - Connectivity

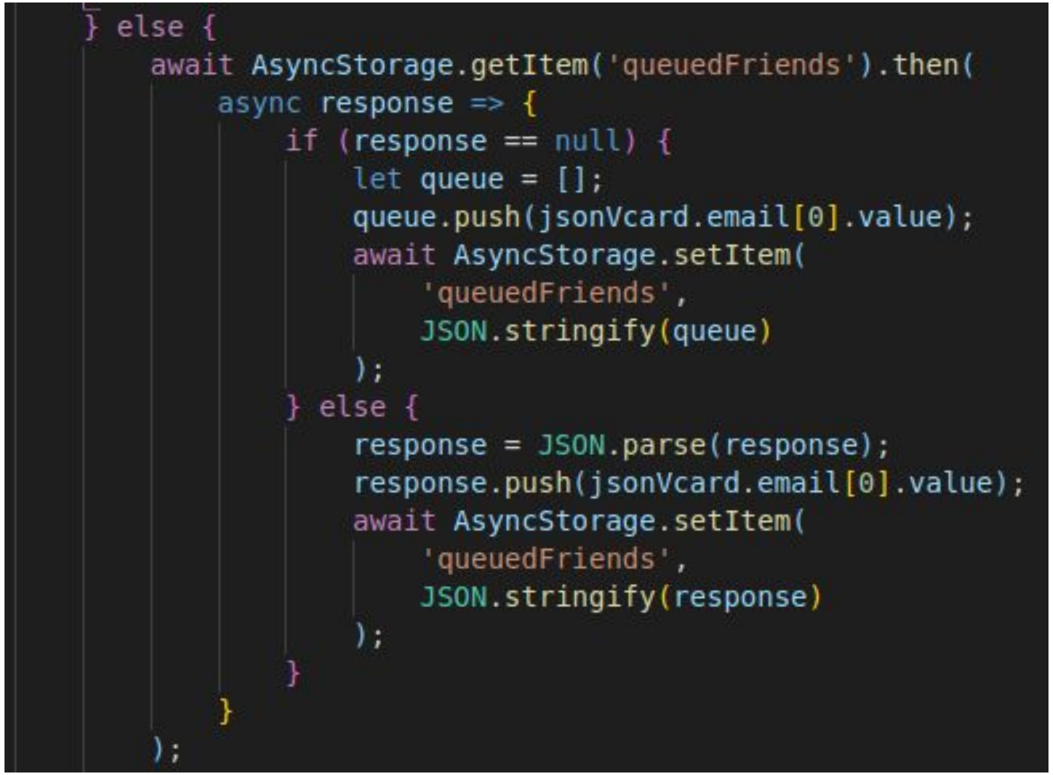
- Cacheable data of the user’s personal information and his friends list including shared information.
- If we have a good connection (good connection = wifi/3G/4G), upload/fetch any changes to personal information/friends to/from server.
- When there is an inexistent or slow connection load data to/from local storage instead and queue any friend additions to be synced the next time the app is online.

Implemented Challenges - Connectivity [Upload/Fetch/Offline check]



```
if (this.state.stableConnection && dirty === 'true') {
    await this.uploadData();
    await AsyncStorage.setItem('dirty', 'false').done();
} else if (
    this.state.stableConnection &&
    (dirty === undefined || dirty === null)
) {
    await this.fetchData();
} else if (!this.state.stableConnection) {
    Toast.show({
        text: 'Offline mode, using cached data',
        type: 'error',
        duration: 2000
    });
}
```

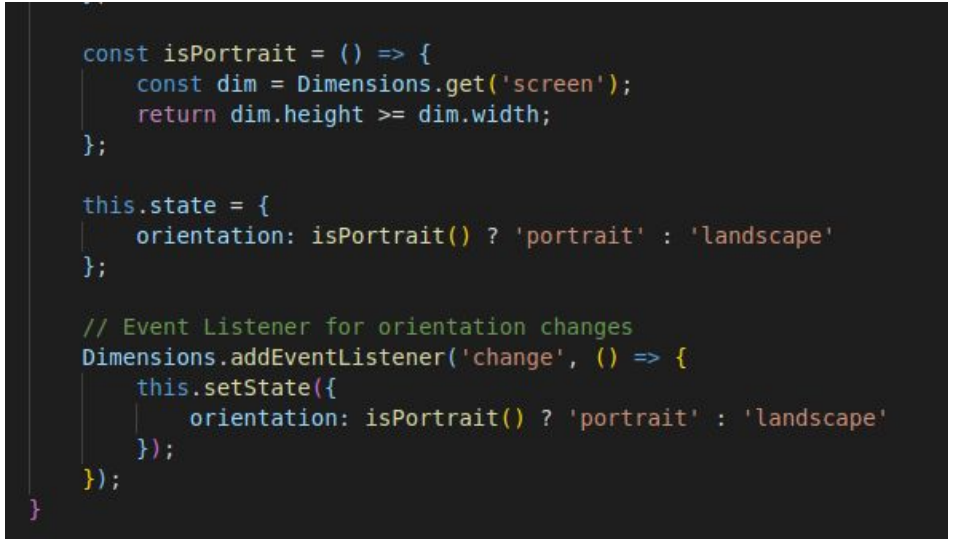
Implemented Challenges – Connectivity [Offline connections queue]



```
    } else {
        await AsyncStorage.getItem('queuedFriends').then(
            async response => {
                if (response == null) {
                    let queue = [];
                    queue.push(jsonVcard.email[0].value);
                    await AsyncStorage.setItem(
                        'queuedFriends',
                        JSON.stringify(queue)
                    );
                } else {
                    response = JSON.parse(response);
                    response.push(jsonVcard.email[0].value);
                    await AsyncStorage.setItem(
                        'queuedFriends',
                        JSON.stringify(response)
                    );
                }
            }
        );
```

Implemented Challenges – Usability

- **Adapting to several screen sizes by using Flex and Dimensions properties.**
- Change UI elements positions to naturally fit landscape mode by using personalized stylesheets.



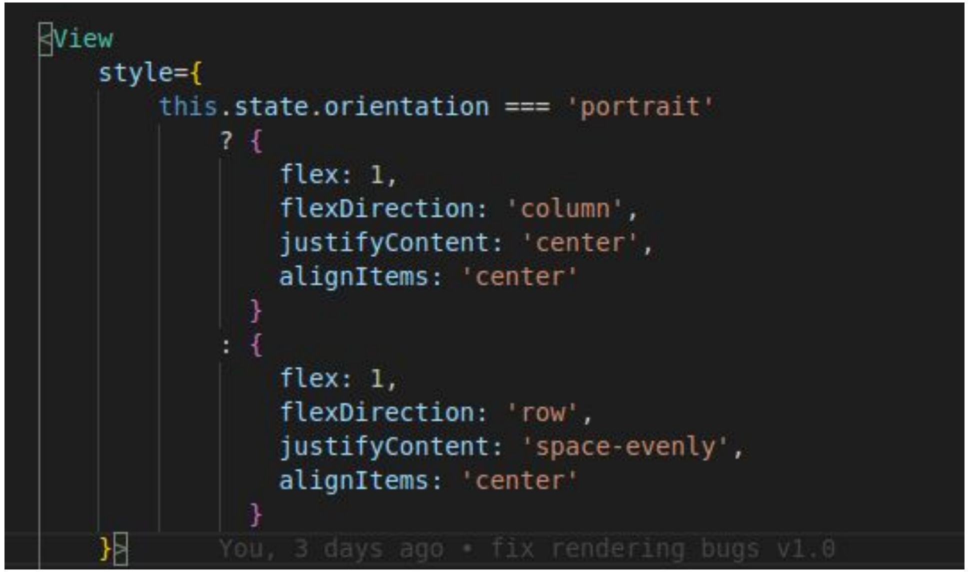
```
    const isPortrait = () => {
        const dim = Dimensions.get('screen');
        return dim.height >= dim.width;
    };

    this.state = {
        orientation: isPortrait() ? 'portrait' : 'landscape'
    };

    // Event Listener for orientation changes
    Dimensions.addEventListener('change', () => {
        this.setState({
            orientation: isPortrait() ? 'portrait' : 'landscape'
        });
    });
}
```

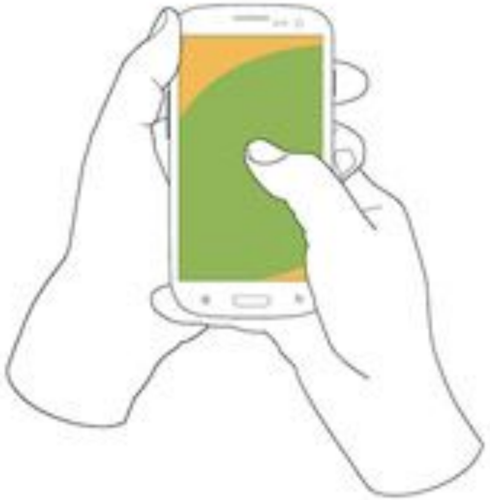
Implemented Challenges – Usability

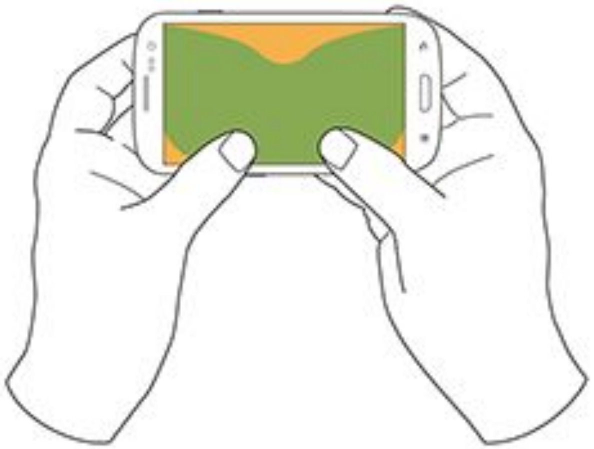
- Adapting to several screen sizes by using Flex and Dimensions properties.
- **Change UI elements positions to naturally fit landscape mode by using personalized stylesheets.**



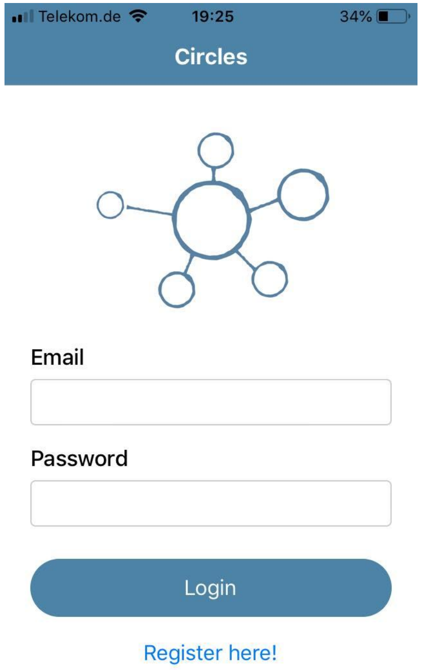
```
<View
    style={
        this.state.orientation === 'portrait'
            ? {
                flex: 1,
                flexDirection: 'column',
                justifyContent: 'center',
                alignItems: 'center'
            }
            : {
                flex: 1,
                flexDirection: 'row',
                justifyContent: 'space-evenly',
                alignItems: 'center'
            }
    }>      You, 3 days ago • fix rendering bugs v1.0
```

Implemented Challenges – Usability [Landscape Mode]





Implemented Challenges - Usability [Landscape Mode]



Telekom.de 19:25 34%

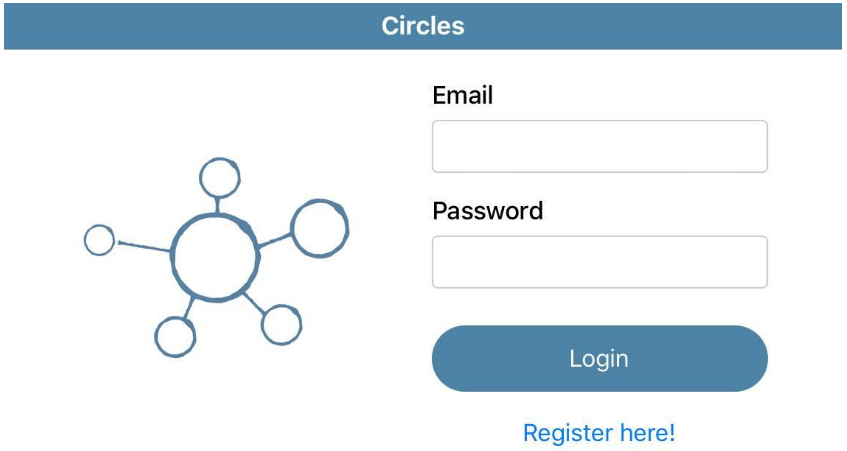
Circles

Email

Password

Login

Register here!



Circles

Email

Password

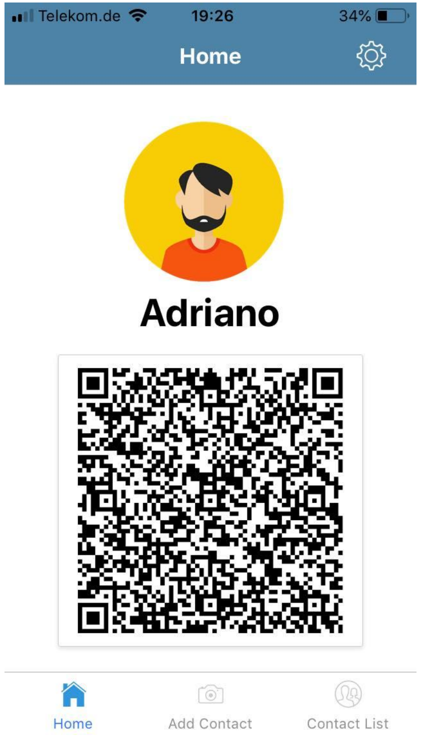
Login

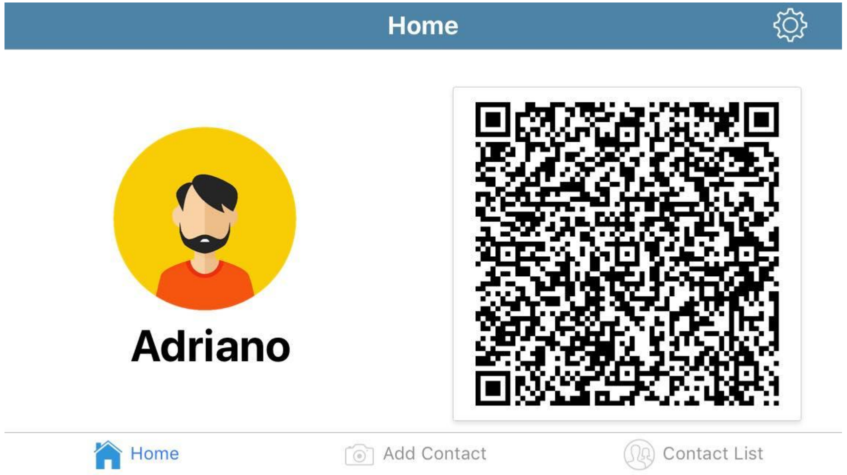
Register here!

TECHNISCHE UNIVERSITÄT DRESDEN

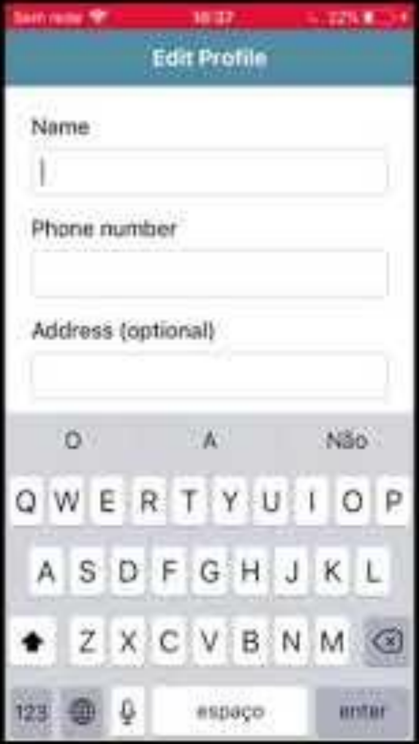
DRESDEN concept

Implemented Challenges – Usability [Landscape Mode]





Register

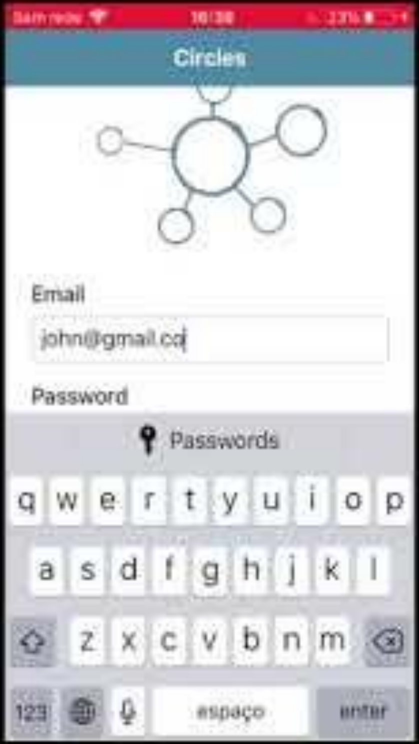

Edit Profile

Name

Phone number

Address (optional)

Login

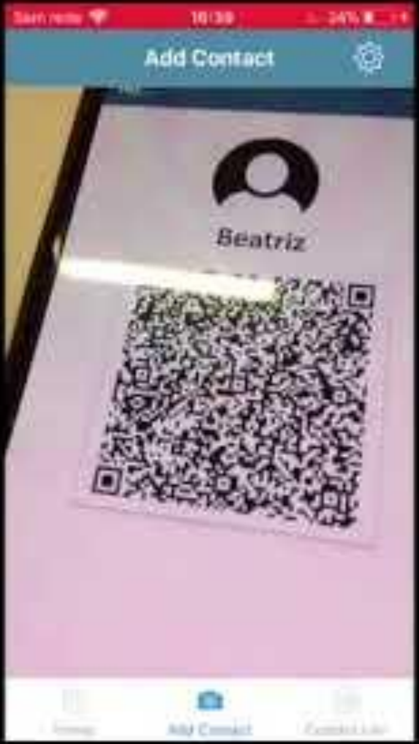

Circles

Email

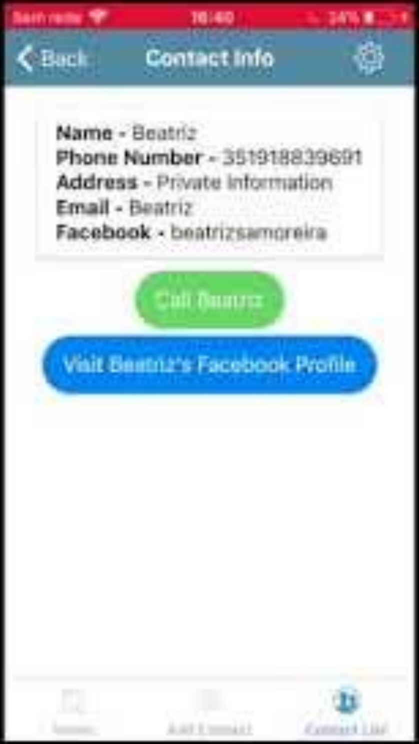
john@gmail.co

Password

Add Contact

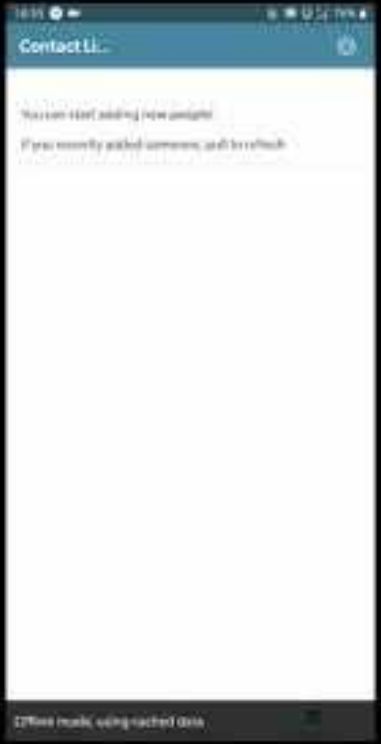


Call Contact


Name - Beatriz
Phone Number - 351918839691
Address - Private Information
Email - Beatriz
Facebook - beatrizsamoreira

Facebook

No Connection



Contact Li...

You can start adding new people!

If you recently added someone, pull to refresh

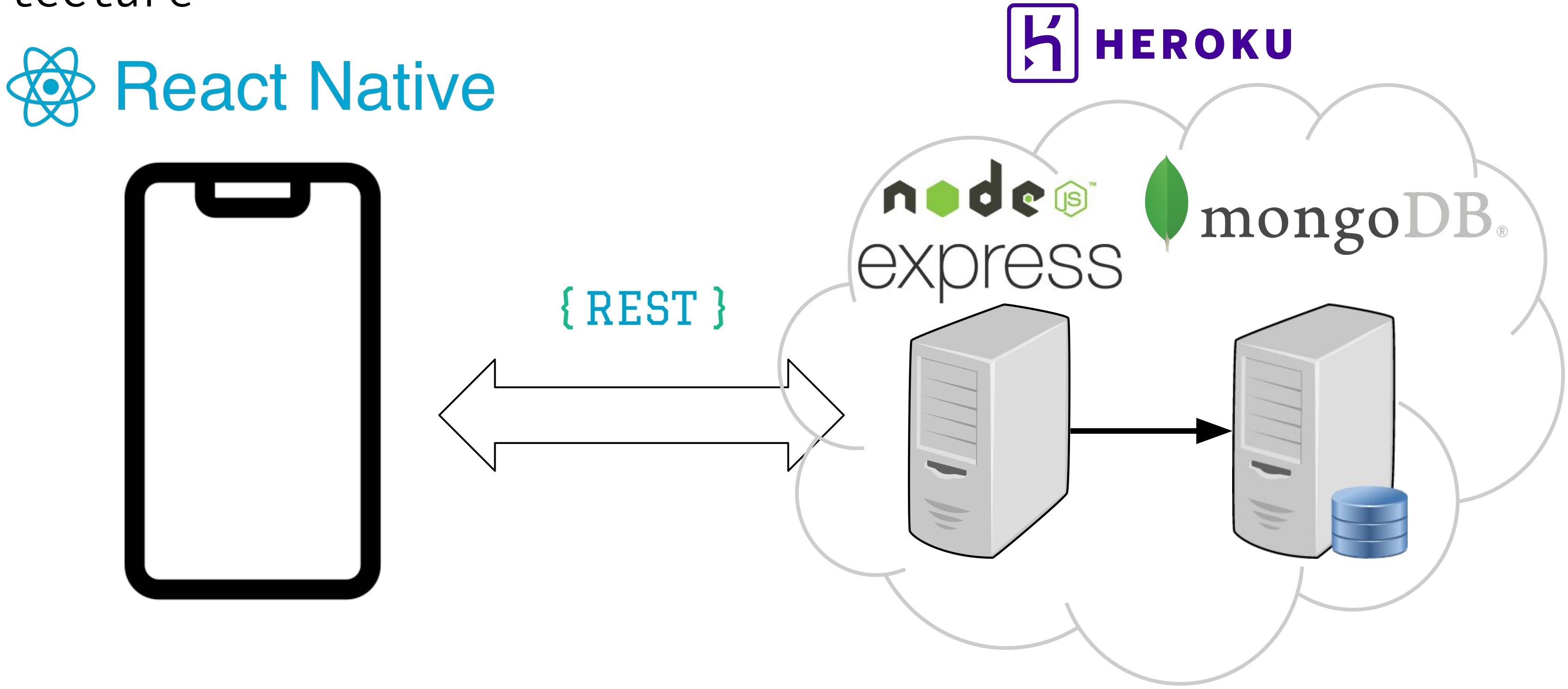
Offline mode, using cached data

Back Online

You can start seeking now [illegible]

If you recently added someone, pull to refresh

Architecture



TODOs & Lessons Learned

- First time developing a fullstack app.
- Putting more effort into designing the DB and REST endpoints pays off.
- Adaptation of UI elements to multiple screen sizes is a painful task.

//TODO:

- You can’t remove friends yet... maybe it’s for the best :)
- Add a maps feature to see the contact’s address
- Further performance improvements
- Small backend improvements