Lernatelier: Projektdokumentation

Andrea Niklaus, Elena Tafelski, Eslem Akdemir, Elisa Petit-Sirigu

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Änderung** | **Autor** |
| 09.11.23 | 0.0.1 | Informieren, Planen, Entscheiden | Petit-Sirigu |
| 16.11.23 | 0.0.2 | Informieren, Planen, Entscheiden | Niklaus |
| 23.11.23 | 0.0.3 | Planen, Entscheiden, Realisieren | Tafelski |
| 30.11.23 | 0.0.4 | Realisieren | Akdemir |
| 07.12.23 | 0.0.5 | Realisieren | Petit-Sirigu |
| 14.12.23 | 0.0.6 | Realisieren, Kontrollieren | Tafelski |
| 21.12.23 | 1.0.0 | Auswerten, Testen | Niklaus, Akdemir |

# Projektantrag zu LA\_ILA3\_0120

|  |  |  |
| --- | --- | --- |
| **Projektteam** | | |
| ***Name*** | ***Vorname*** | ***Klasse*** |
| Petit-Sirigu | Elisa | IM21b |
| Tafelski | Elena | IM21d |
| Niklaus | Andrea | IM21d |
| Akdemir | Eslem | IM21b |
| **Projektidee / Produkt** | | |
| QR-Code Scanner Mobile App mit Expo | | |
| **Betroffene Informatik-Module** | | |
| Modul 335 | | |
| **Infrastruktur** | | |
| *Wie organisieren Sie Ihre Zusammenarbeit? Wo ist die Ablage für den Code? Wo wird die Dokumentation abgelegt?*    Wir arbeiten in einem vierer Team und werden unsere Aufgaben wie folgt aufteilen:  Jeder übernimmt einen Teil der Anforderungen, die wir in der Dokumentation festhalten werden. Dabei arbeitet jeder auf einem eigenen Branch, die wir am Schluss in ein dev Branch mergen werden. Auf dem dev Branch wird getestet. Recherchieren werden wir alle und das Dokumentieren werden wir auch untereinander aufteilen.   Wichtige Codestellen, wie zum Beispiel das Scannen vom QR-Code werden wir zusammen machen, weil jeder von uns wissen will, wie genau es funktioniert.    Ablage Code: Github  Dokumentation: Per Sharepoint, Schluss vom Projekt in Github hochladen | | |
| **Projektmethode** | | |
| Wir gehen nach IPERKA vor. | | |
| **Persönliche Ziele** | | |
| Akdemir, Eslem (abwesend am ersten Tag)  Ziel 1: Ich weiss, wie man ein Visual Studio Code mit dem Handy verbindet.  Ziel 2: Ich weiss, wie man einer mobilen App Kamerazugriff gibt.  Ziel 3: Ich weiss, wie man eine App programmiert, die QR-Codes entschlüsseln kann | | |
| Niklaus, Andrea (teilweise abwesend am ersten Tag)    Ziel 1: Ich lerne, wie ein Sensor von Expo zu benutzen  Ziel 2: Ich weiss, wie die Mobile-App designen und wie die Navigation zwischen dem Screen Home und dem QR-Code Scanner läuft und kann dies auch umsetzten.  Ziel 3: Ich weiss, wie der Aufbau von Expo Kamera/Expo QR-Code funktioniert und wie ich dies in die Applikation erfolgreich implementieren kann. | | |
| Tafelski, Elena    Ziel 1: Ich weiss, wie man einen QR-Code entschlüsselt, indem wir einen QR-Code-Scanner programmieren.  Ziel 2: Ich kann mit Expo eine Mobile App programmieren, nachdem wir das in diesem Projekt gemacht haben.  Ziel 3: Ich weiss, wie die Kamera in einer Mobile App verwendet werden kann, indem wir die Kamera benutzen, um den QR-Code zu scannen. | | |
| Petit-Sirigu Elisa    Ziel 1: Ich weiss, wie ich einen QR-Code Scanner programmiere, indem ich den QR-Code Scanner mit den anderen programmieren werde.    Ziel 2: Ich kann Objekte in der App besser Platzieren und designen, indem ich mich darüber informieren und diese Aufgabe im Projekt übernehme.    Ziel 3: Ich weiss, wie ich ein Zugriff auf die Kamera programmieren kann, indem ich den Zugriff mit den anderen programmieren werde. | | |

# Informieren

## Ihr Projekt

In diesem Projekt wollen wir mir Expo eine App erstellen, die einen QR-Code scannen kann. Diesen Scanner wollen wir selbst programmieren. Die App soll, wenn wir noch genügend Zeit dafür haben, ein schönes Design haben: Keine schwarz-weisse App, sondern ein Userinterface mit Farben und Buttons.

**Der Link zum Projekt auf Github ist folgender:** <https://github.com/titepasile/QR-CodeScanner>

## Quellen

[BarCodeScanner - Expo Documentation](https://docs.expo.dev/versions/latest/sdk/bar-code-scanner/)

<https://www.dynamsoft.com/codepool/html5-barcode-reader-javascript-webassembly.html> (nicht für Expo)

<https://github.com/mebjas/html5-qrcode> (nicht für Expo)

<https://tech-market.org/react-native-qr-scanner/>

<https://qiita.com/hitotch/items/de06a44b3aa699f0f4cb>

[medium-qr-code-scanner](https://medium.com/@Sarmilasivaraja/scan-qr-code-with-expo-camera-react-native-c60be39fdb7d)

<https://blog.minhazav.dev/QR-and-barcode-scanner-using-html-and-javascript/>

<https://www.youtube.com/watch?v=3mMyd3r2LRc&ab_channel=BugNinza>

<https://reactnavigation.org/>

<https://www.freecodecamp.org/news/how-to-create-a-camera-app-with-expo-and-react-native/>

<https://colorhunt.co/palette/f1ddbf525e7578938a92ba92> (für Farbpalette)

cmd für Kamera: expo install expo-camera

Folgendes Logo für die App selbst gemacht (von Andrea Niklaus):

Ein Bild, das Design, Grafiken, Symbol, Rechteck enthält.

Automatisch generierte Beschreibung

## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
|  | MUSS | Funktional | Unsere App soll einen QR-Code scannen können |
|  | MUSS | Funktional | Die App soll das Normalformat eines QR-Codes scannen können |
|  | MUSS | Qualität | Nach dem erfolgreichen Scannen erscheint eine Bestätigung |
|  | MUSS | Qualität | Auf der Bestätigung steht, dass der Scann erfolgreich war. |
|  | MUSS | Qualität | Nach dem unerfolgreichen Scannen erscheint eine Fehlermeldung |
|  | MUSS | Qualität | In der Fehlermeldung steht, dass der Scann Vorgang wiederholt werden soll. |
|  | MUSS | Funktional | Die App muss einen Scan Button haben |
|  | MUSS | Funktional | Der Scan Button soll das Scannen vom QR-Code starten |
|  | KANN | Qualität | Die App soll ein Logo haben |
|  | MUSS | Rand | Die App soll Scanner heissen |
|  | KANN | Qualität | Die App soll ein schönes Design haben. Kein Schwarz-Weiss |
|  | MUSS | Funktional | Die App soll Zugriff auf die Kamera haben |
|  | MUSS | Rand | Der Code soll in JavaScript programmiert sein |
|  | MUSS | Rand | Wir werden mit Visual Studio arbeiten |
|  | MUSS | Rand | Unsere App soll mit Expo erstellt werden |

## Testfälle

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Voraussetzung** | **Eingabe** | **Erwartete Ausgabe** |
|  | Die App ist geöffnet | Scanner aktivieren | QR-Code wird erfolgreich gescannt |
|  | Die App ist geöffnet und der Scanner gestartet | Einen quadratischen QR-Code scannen der 20 x 20 mm gross ist. | Scanne hat funktioniert. |
| 2.2 | Die App ist geöffnet und der Scanner gestartet | Einen QR-Code einer nicht als Norm angesehener Grösse scannen. (alles andere als 20 x 20 mm) | Scannen hat nicht funktioniert |
|  | Der Scanner wurde gestartet | Der QR-Code wurde erfolgreich gescannt. | Eine Meldung erscheint |
|  | Der QR- Code wurde erfolgreich gescannt und die Meldung ist erschienen. | Keine Eingaben notwendig | Auf der Meldung steht, dass das Scannen erfolgreich war. |
|  | Der Scanner wurde gestartet | Der QR-Code konnte nicht gescannt werden. | Eine Meldung erscheint. |
|  | Der QR- Code konnte nicht gescannt werden und die Meldung ist erschienen. | Keine Eingaben notwendig | Auf der Meldung steht, dass das Scannen nicht erfolgreich war. |
|  | Die App ist geöffnet und ein Scan Button ist zu sehen. | Auf den Scann Button klicken | Die Kamera öffnet sich. |
|  | Die App ist geöffnet, ein QR-Code vorhanden und ein Scan Button ist zu sehen. | Auf den Scann Button klicken | Die Kamera öffnet sich und der QR-Code wird gescannt. |
|  | Man befindet sich in der App | Die App schliessen | Die App hat ein Logo und nicht mehr das normale placeholder Bild. |
|  | Man befindet sich in der App | Die App schliessen | Die App heisst Scanner |
|  | Man befindet sich nicht in der App | Die App öffnen | Das Userinterface hat ein schönes design mit Farben und ist nicht schwarz-weiss. |
| 1.12 | Die App ist geöffnet | Auf den Scann Button klicken | Die Kamera öffnet sich --> die kann verwendet werden. |

# Planen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** | **Hauptverantwortliche** |
| 0 | 16.11.23 | Projektdokumentation beginnen IPE | 200 | alle |
| 1 | 16.11.23 | Die App existiert und kann geöffnet werden | 50 | Petit-Sirigu |
| 2 | 30.11.23 | Der Scanner soll nur einen quadratischen QR-Code scannen | 60 | Petit-Sirigu |
| 3 | 30.11.23 | Zu grosser QR-Code darf nicht eingescannt werden | 60 | Niklaus |
| 4 | 07.12.23 | Eine Meldung kommt nach einem erfolgreichen Scan | 180 | Tafelski |
| 5 | 07.12.23 | Der QR soll entschlüsselt werden | 200 | Petit-Sirigu |
| 6 | 07.12.23 | Nach dem Entschlüsseln vom QR-Code soll der Link/Text in einem Textfeld dem Benutzer gezeigt werden | 200 | Niklaus |
| 7 | 30.11.23 | Der Scanbereich soll 20x20 mm sein | 60 | Petit-Sirigu |
| 8 | 14.12.23 | Eine Fehlermeldung erscheint bei einem nicht erfolgreichen Scan | 120 | Petit-Sirigu |
| 9 | 30.11.23 | Kamera erscheint nach dem Klick auf den Scan Button | 180 | Tafelski |
| 10 | 14.12.23 | Appdesigns (Siehe Entscheiden) à Farben, Layout | 90 | Niklaus |
| 11 | 14.12.23 | Userinterface ist ersichtlich und klar bedienbar | 20 | Tafelski |
| 12 | 23.11.23 | Titel der App ersichtlich | 40 | Tafelski |
| 13 | 23.11.23 | Logo der App erstellen | 80 | Niklaus |
| 14 | 23.11.23 | Logo der App ersichtlich | 40 | Niklaus |
| 15 | 23.11.23 | Zugriff auf Kamera | 150 | Niklaus |
| 16 | 14.12.23 | Testen und Testprotokoll mit Testumgebung und Fazit schreiben | 20 | Akdemir |
| 17 | 21.12.23 | Portfolioeintrag schreiben | 220 | alle |

# Entscheiden

Welches Projekt wollen wir machen?

In der engeren Auswahl standen ein Calculator und ein QR-Code Scanner. Wir haben uns schlussendlich für den QR-Code Scanner entschieden, weil wir einen Sensor in die App integrieren wollten. In diesem Fall ist es der Zugriff auf die Kamera.

### Welches Design wollen wir für unsere App (Farben)?

* Pastellfarben
* Einheitliche Farben

### Wie soll der Button aussehen?

* Einheitlich mit dem Hintergrund der App

à Sollte jedoch erkennbar sein

* Rundliche Ecken

### Wie soll unseres App-Logo aussehen?

* Selbstgemachtes Logo
* Simples Logo wie Piktogramme aus Word
* Passend zur App

# Realisieren

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr.** | **Frist** | **Beschreibung** | **Zeit (geplant)** | **Zeit (effektiv)** | **Hauptverantwortliche** |
| 0 | 16.11.23 | Projektdokumentation beginnen IPE | 200 | 200 | alle |
| 1 | 16.11.23 | Die App existiert und kann geöffnet werden | 50 | 40 | Petit-Sirigu |
| 2 | 30.11.23 | Der Scanner soll nur einen quadratischen QR-Code scannen | 60 | 20 | Petit-Sirigu |
| 3 | 30.11.23 | Zu grosser QR-Code darf nicht eingescannt werden | 60 | 70 | Niklaus |
| 4 | 07.12.23 | Eine Meldung kommt nach einem erfolgreichen Scan | 180 | 100 | Tafelski |
| 5 | 07.12.23 | Der QR soll entschlüsselt werden | 200 | 15 | Petit-Sirigu |
| 6 | 07.12.23 | Nach dem Entschlüsseln vom QR-Code soll der Link/Text in einem Textfeld dem Benutzer gezeigt werden | 200 | 130 | Niklaus |
| 7 | 30.11.23 | Der Scanbereich soll 20x20 mm sein | 60 | 10 | Petit-Sirigu |
| 8 | 14.12.23 | Eine Fehlermeldung erscheint bei einem nicht erfolgreichen Scan | 120 | 30 | Petit-Sirigu |
| 9 | 30.11.23 | Kamera erscheint nach dem Klick auf den Scan Button | 180 | 160 | Tafelski |
| 10 | 14.12.23 | Appdesigns (Siehe Entscheiden) à Farben, Layout | 90 | 80 | Niklaus |
| 11 | 14.12.23 | Userinterface ist ersichtlich und klar bedienbar | 20 | 25 | Tafelski |
| 12 | 23.11.23 | Titel der App ersichtlich | 40 | 40 | Tafelski |
| 13 | 23.11.23 | Logo der App erstellen | 80 | 80 | Akdemir |
| 14 | 23.11.23 | Logo der App ersichtlich | 40 | 20 | Niklaus |
| 15 | 23.11.23 | Zugriff auf Kamera | 150 | 130 | Niklaus |
| 16 | 14.12.23 | Testen und Testprotokoll mit Testumgebung und Fazit schreiben | 20 | 20 | Akdemir |
| 17 | 21.12.23 | Portfolioeintrag schreiben | 220 | 220 | alle |

# Kontrollieren

## Testprotokoll

**Testumgebung: Expo Go, Visual Studio Code**  
**Computer/Handy: Iphone 13**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Test Nr.** | **Datum** | **Resultat** | **Durchgeführt** |
|  | 1.1 | 21.12.23 | **OK** | Akdemir |
|  | 1.2 | 21.12.23 | **OK** | Akdemir |
|  | 1.3 | 21.12.23 | **OK** | Akdemir |
|  | 1.4 | 21.12.23 | **OK** | Akdemir |
|  | 1.5 | 21.12.23 | **OK** | Akdemir |
|  | 1.6 | 21.12.23 | **OK** | Akdemir |
|  | 1.7 | 21.12.23 | **OK** | Akdemir |
|  | 1.8 | 21.12.23 | **OK** | Akdemir |
|  | 1.9 | 21.12.23 | **OK** | Akdemir |
|  | 1.10 | 21.12.23 | **OK** | Akdemir |
|  | 1.11 | 21.12.23 | **OK** | Akdemir |
|  | 2.11 | 21.12.23 | **OK** | Akdemir |

**Fazit: Alle Testfälle wurden erreicht, die App ist ganz funktionsfähig.**

## Auswerten

Über das Projekt:

Das Projekt verlief sehr gut. Wir waren immer gut im Zeitplan, hatten keinen Stress, obwohl viele im Team einmal krank waren, und wir waren in der Lage, alle Anforderungen zu erfüllen. Beim Entscheiden der Projektthemas waren wir uns Anfangs nicht sehr schlüssig. Allerdings konnten wir alle anderen Entscheidungen, wie zum Beispiel das Wählen der Farben für unser Design, sehr schnell treffen.   
Das Implementieren des QR-Code Scanners war einfacher als gedacht, da man dafür den Expo Barcode Scanner importieren kann.

Über das Team:

Die Kommunikation und Arbeitseinteilung waren eher schwierig, da wir uns nicht gut kannten und niemand so richtig die Verantwortung für das Projekt übernehmen wollte. Wegen den vielen Ausfällen und dem Onlineunterricht blieb das auch leider so.

Jedoch konnten wir gut zusammenarbeiten und hatten keine Konflikte in der Gruppe. Dadurch sind wir auch sehr schnell vorangekommen.