

20.11.2019 // Open Access

System_Dokument zum Verbundtreffen

Du möchtest Teil des Projektes werden? Melde dich bei Max

Hier geht es zum aktuellen Projekt-Dokument:

<https://pad.gwdg.de/DsyvzN4TQyyB5M94ZCBihQ?view>

Kurzvorstellung des Teilprojektes FabAccess

FabAccess ist ein Zugangssystem, das, basierend auf den verschiedenen Ansätzen und gemachten Erfahrungen der einzelnen Werkstätten, eine möglichst breite Nutzbarkeit ermöglichen soll. Ziel ist die Entwicklung möglichst objektbasierter und generischer Schnittstellen, die die diversen Bedürfnisse der Werkstätten abdecken.

Hier der Link zum aktuellen Clickt-Dummy des Admininterfaces - ein **Gitlab-Rep folgt**:

<https://fabaccess.bian-meyer.de/>

Authentifizierungsmethoden: 13,56Mhz RFID-Karten und Pin, QR-Code über Nutzer-App

Specs:

- Aktoren werden aktuell angesprochen über openHAB - weitere Schnittstellen sind geplant
- Schnittstelle zu und Datenbankbasis über Odoo
- Odoo ermöglicht die Rechnungstellung / Abrechnungen von Aktionen
- Schaltbar gemacht werden sollen diverse Aktoren, wie Türen, große/kleine Maschinen etc.
- Individuelle und maschinenbaiserte Vergabe von Berechtigungen (Person X hat eine Einführung Y bekommen und damit Zugang zu Z oder als Mitglied Zutritt zu der Tür A)

Doku zum Workshop auf dem Verbundtreffen der Offenen Werkstätten 2019

Das Projekt sowie der aktuelle Stand wurde vorgestellt. Im Anschluss folgt das Feedback der Teilnehmenden:

- individuelle Verbrauchswerte sollten erfasst werden können
- es sollte eine einfache Möglichkeit der Stornierung geben, wenn eine Falschbuchung gemacht wurde
- der Preis der aktuellen Verbrauchszeit sollte dem User angezeigt werden
- es sollte unterschiedliche Typen von "Rechnungen" geben. Nicht jeder Nutzer braucht eine volle Rechnung mit mwst.
- Verschleißdaten / Nutzungsdaten von Maschinen sollten angezeigt werden können: Rohdaten!
- Nutzungsevent wird als Nachricht formuliert, die andere Systeme konsummieren können.
- Stammdaten nicht als Pflichtfelder anlegen
- Berechtigungen suchen
- Benutzergruppen erzeugen? Wäre einfacher, wenn vielen die Berechtigung für etwas erteilt werden soll
- mehrere Zugriffsgeräte für einen Ort, um die dort stehenden Geräte zu schalten
- importieren / exportieren von Geräten
- die Möglichkeit Fotos zu hinterlegen für "Orte", um dort Geräte wie in "real" anordnen zu können
- die Möglichkeit der "Rückkommunikation" schaffen
- Adminprofil, das die App freischaltet, um sie zu konfigurieren

Folgende Fragen wurden gegen Ende gestellt:

- Könnt ihr damit eure Nutzerfälle abbilden?
 - Es kam kein besonderes negatives Feedback
- Welche Authentifizierungsmethoden nutzt ihr?
 - nfc
- Welche speziellen Aktoren sollen genutzt werden?
 - Drehstromaschinen ermöglichen

Wünsche / To-Dos:

- offenes git erstellen

- Webapp ermöglichen
- Nutzer abmelden auf Tablets
- Zeiträume für Aktoren / Maschinen festlegen
- wenn Signal eines Aktors abbricht, Buchung nach einer Zeit beenden (z.b. im Falle eines Notaus)
- 3 Use-Cases entwickeln, die möglichst breite Anwendungsfälle abbilden
 - Boxen für Kleingeräte, statt das Kleingerät selbst zu schalten
 - ...
 - ...

Digitalisierung der Klebezettel, die von den Teilnehmern beschrieben wurden

- Anbindung eigener Nutzerdatenbank (anstatt Odoo)
- Tablet: aktives Ausschalten realisieren???
- Rollenprofile
 - Admin Zugangskarte
 - Einweiser Karte
- Schließfächer, Zigarettensautomat, Vendingmaschine???
- Anbindung von Automationshardware der Firma Pilz (<https://www.pilz.com/de-DE/produkte-loesungen/schaltgeraete>)
- Vermerken, wann eine und von wem eine Berechtigung erteilt wurde
- Manuelle Verbrauchserfassung
- Kontextinformationen zur Buchung hinzufügbarmachen (frei)
- Anbindung/Monitoring durch Messages & Events
- Ein Zugriffsgerät mehreren Orten zuweisen können
- Benutzern gesammelt Berechtigungen erteilen
- Verschiedene Benutzergruppen (mit und ohne Rechnung)
- Korrekturbuchungen möglich machen (Rechnung)
- Anbindung von Codesys
- Verschiedene Admins (1 Admin für Vereine, 1 Admin für Vereine)

Technologien/Tools die Werkstätten verwenden

- **Fablab Chemnitz (nacheditiert am 20.11.2019)**
 1. **was verstehen wir unter einem Zugangssystem?**
 - “Hardware as a Service” (HaaS) in Anlehnung zu “Software as a Service” (SaaS)

- Ressourcenverwaltung / Inventarverwaltung
 - Nutzerverwaltung / Mitgliederverwaltung
 - Problem: für verschiedene Aufgaben haben wir bereits erprobte und gut genutzte Software. Wie verdrahten wir das, was wir bereits haben mit dem, was noch fehlt? Permanente Migration zu anderen Programmen ist zu aufwendig und zu fehleranfällig
-

2. **konkret vorhandene Hardware, die wir einbinden wollen und bereits besitzen**

- Nuki Smartlock mit Nuki Bridge
 - <https://nuki.io/de/>
 - mehrere Devolo Home Control Steckdosen (außerdem Temperatursensor, Rauchmelder, etc. vom gleichen Anbieter)
 - <https://www.devolo.de/devolo-home-control-schalt-und-messsteckdose-20>
 - Tinkerforge NFC Card Reader Terminal (hier sollen sich unsere Mitglieder später mit einem NFC-Tag an- und abmelden können bzw. weitere Aktionen ausführen)
 - <https://www.nfc-tag-shop.de/sale/nfc-anhaenger-40-x-32-mm-topaz-512-512-byte-blau-229>
 - <https://www.tinkerforge.com/de/doc/Hardware/Bricklets/NFC.html>
 - NeoCoolCam Power Plugs
 - <https://www.amazon.de/Coolcam-Z-Wave-Smart-Metering-Packung/dp/B072XSBKJT>
 - langfristig wollen wir neben Einzelgeräten unsere Hauselektronik über Hutschienenelektronik steuerbar machen, also direkt über die (Unter)verteilerkästen -> erlaubt intelligentes Zuschalten von Werkstattbeleuchtung o.ä.
-

3. **Wünsche an "FabAccess"**

- Software sollte von Sensoren möglichst viele Daten speichern (z.B. in einer Zeitdatenbank wie InfluxDB) - z.B. Stromverlauf, Einschalten, Ausschalten, Widerstandswerte, etc - "alles, was die Sensoren hergeben" -> Anzeigen + Auswerten von Lastgraphen hilft dabei ein Action-Event-Trigger-basiertes System zu entwickeln
- Einbindung an SingleSignOn (SSO), z.B. LDAP oder Radius (wir wollen irgendwann mal OpenLDAP und PrivacyID3A 2FA verwenden)

- Template-Customizing -> Farben, Banner, Icons anpassbar machen
 - Berechtigungen von Benutzern sollten zeitlich konfigurierbar sein (Benutzer darf an Tagen X,Y,Z in den Zeitfenstern A,B,C) -> Zeitfenster schalten für Gäste, mobile Maker, Workshop-Teilnehmer, etc.
 - ggf. Plugin-Architektur
 - System mit i18n - Basics: Deutsch + Englisch
 - Import/Export/Backup-Funktionen (ggf. Rollback von Konfigurationen bei fehlerhaften Settings) > XML, JSON
 - Installierbarkeit mit oder ohne Containerisierung (Docker) auf wichtigen Systemen wie Linux-Distributionen, Windows, (MAC?)
 - ggf. Reservieren/Ausleihen (ausbuchen) von mobilem Equipment
 - Custom Property Fields zum Befüllen durch den Admin: bei den Reitern "Maschinen" und "Orte" wäre es schön, wenn man mehrere externe URLs anbinden bzw. Infos anreichern kann, die den Nutzer zu spezifischen Unterseiten verlinkt (z.B. externe Handbücher, Cloud-URLs, Fotos, etc.) oder z.B. diverse Eigenschaften/Kommentare anzeigt
-

4. Infos bezüglich Software, die wir im FabLab Chemnitz verwenden >> Interoperabilität wünschenswert

- Teedy -> Dokumentenmanagement (Langzeitarchivierung Rechnungen, etc.)
 - JTL Shop+WaWi -> speichern von Mitgliederstammdaten und Erfassen von Nutzungsgebühren (JTL Shop Anbindung)
 - ZDF WISO MeinVerein - Rechnungslegung, Buchhaltung
 - Grafana -> Monitoring, grafische Auswertung von Datenbanken (PostgreSQL, MariaDB, MySQL, MSSQL)
 - Repetier Server (3D-Druck)
 - Octoprint Server (3D-Druck)
 - Confluence (Dokumentation zu Inventar, Tutorials, Projekte, IT-Stuff)
 - Matrix Synapse (offener Chat-Server m. Bot-Integrationen) -> könnte später mal Statuswerte aus FabAccess anzeigen
 - sonstiger Web-Kram im Peripheriebezug: WordPress, Plesk, PicApport, sysPass, Gitea
 - diverse (bash-)scripts mit API Calls, die unsere Systeme bereits etwas verdrahtet
-

5. Allgemeine Gedanken/Fragen zu FabAccess

- Ausfallsicherheit bedenken. Wie funktionieren die Vorgänge, wenn mal etwas offline ist? > Flexibilität in der manuellen Nacharbeit notwendig, falls der Server mal ein paar Tage nicht geht!

- welche Lizenz soll das Projekt haben? z.B. MIT, BSD, Apache, GNU GPL?
- unser damaliger Gedanke zu Abrechnung, Zugang, allgemeiner Verwaltung:

<https://wiki.fablabchemnitz.de/display/FCP/Projekt+Farringdon+ins+Leben+gerufen!+Die+zentrale+Inventar-+und+Buchungsplattform>

- **Munich Makerlab**

- Selbstgebautes System zum öffnen von Türen
-

Version #2

Erstellt: 13 Oktober 2024 01:16:03 von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 14 Dezember 2024 18:23:07 von Mario Voigt (Stadtfabrikanten e.V.)