Odisee
DE CO-HOGESCHOOL

EduBadge

HET BADGESYSTEEM OM AANWEZIGHEDEN TE REGISTREREN

Mauro De Bisschop

Sander De Winne

Jethro Meerts

Wout Rommel

Elektronica-ICT

2024 - 2025

Mentor: Joris Maervoet, Sabine Martens

Opdrachtgever: Odisee opleiding Energietechnologie

EduBadge: An RFID-Based Attendance Registration System

M. De Bisschop, S. De Winne, J. Meerts, W. Rommel

This paper introduces *EduBadge*, an RFID-based attendance registration system designed to improve attendance tracking in the Odisee Energy Technology labs. Traditional manual attendance methods are time-consuming and prone to errors, increasing the administrative workload for instructors. EduBadge aims to automate this process, reducing manual effort and improving accuracy.

The system utilizes RFID-technology, allowing students to register their presence by scanning their student cards or temporary badges. For students without a student card, a badge dispenser issues temporary RFID-badges linked to their accounts. The hardware setup includes an RFID-reader for identification, an ESP32 microcontroller responsible for controlling the badge dispenser and managing communication with a centralized database and a TFT-touchscreen display enabling user interaction and account linking.

On the software side, EduBadge features a backend system connected to a centralized database that stores user profiles, attendance records, and system settings across multiple tables. The backend handles authentication via Time-based One-Time Passwords (TOTP), e-mail verification for account activation and secure communication with the hardware using HMAC tokens. It also provides data filtering and sorting to ensure efficient and secure data management.

The frontend is implemented as a browser-based user interface using TypeScript, offering a dynamic navigation bar that adapts to the user's login status and role. Asynchronous HTTP requests allow smooth data exchange with the backend. Both client-side and server-side session management control access and tailor the interface dynamically. Input validation is performed on the client side to maintain data integrity and a contact form is integrated using Formspree.

Security is a fundamental aspect of EduBadge, with encrypted communications, robust authentication and role-based access control applied throughout the system. Initial testing confirms that EduBadge successfully reduces administrative overhead while improving attendance accuracy and user convenience.

Keywords: RFID, attendance tracking, ESP32, TOTP-authentication, web application, security.

Inhoudsopgave

Figurenlijst

Tabellenlijst

Codefragmentenlijst

Inleiding

In dit rapport wordt stap voor stap de opbouw en werking van EduBadge beschreven, een registratiesysteem dat gebruikmaakt van badges. Binnen de labo's van Energietechnologie aan Odisee is het bijhouden van aanwezigheden een vereiste. Met dit systeem kunnen studenten zichzelf efficiënt inchecken, waardoor het registratieproces sneller en eenvoudiger verloopt.

Momenteel worden de aanwezigheden in de labo's handmatig bijgehouden. Dit is niet alleen een administratieve last, maar ook tijdrovend voor de docenten. Om dit te optimaliseren, werd de vraag gesteld naar een registratie- en loggingsysteem dat onderhoudsvrij is en zo min mogelijk afhankelijk van externe systemen. De registratie van studenten zou plaatsvinden via een RFID-lezer, wat zorgt voor een efficiënte en geautomatiseerde verwerking van aanwezigheden. Hoewel er geen vast budget is opgesteld voor dit systeem, wordt er wel verwacht dat de kosten zo laag mogelijk blijven.

Het systeem maakt gebruik van een externe badgelezer, waarbij studenten hun studentenkaart als badge kunnen gebruiken nadat ze deze gekoppeld hebben aan hun account. Voor studenten die hun studentenkaart (nog) niet hebben, moet er een systeem worden voorzien waarmee ze tijdelijke badges kunnen verkrijgen. Een automaat lijkt hier passend aangezien deze makkelijk badges kan uitgeven.

Daarnaast wordt een webpagina aangeboden waarop studenten hun aanwezigheden kunnen raadplegen en hun badge kunnen beheren, bijvoorbeeld door deze in te schakelen of te vervangen bij verlies. Voor docenten biedt de website een overzichtelijke interface waarmee ze de aanwezigheidsregistraties per lokaal of per vak kunnen controleren.

Beveiliging is een belangrijk aspect van het systeem. De website moet beveiligd worden zodat niet iedereen zomaar aan alle data kan. Een idee is om met een login te werken die via een externe verificatie werkt. Op deze manier wordt de toegang van ongeautoriseerde gebruikers voorkomen. In de database wordt de naam van de gebruiker, de RFID-tag en het e-mailadres opgeslagen. Deze gegevens moeten goed beveiligd worden en enkel de informatie die de gebruiker mag zien wordt via de website toegankelijk.

Voor de opmaak van dit rapport werd gebruikgemaakt van AI-hulpmiddelen ter ondersteuning van de tekstuele uitwerking[1].

In het eerste en tweede hoofdstuk wordt de hardware van het project besproken, inclusief de voor- en nadelen van de gekozen apparatuur en hoe deze geschakeld is. Het derde hoofdstuk richt zich op de software, met nadruk op de backendwerking van het systeem. In het vierde hoofdstuk worden de securitymaatregelen die in het systeem zijn geïmplementeerd behandeld. Vervolgens wordt in het vijfde hoofdstuk het ontwerp van de automaat en de scanner besproken, alsook het ontwerp van de printplaten. In het zesde hoofdstuk wordt de totale kostprijs van het project berekend. Tot slot beantwoordt de conclusie of er aan de eisen voldaan wordt en worden mogelijke toekomstige verbeterpunten besproken.

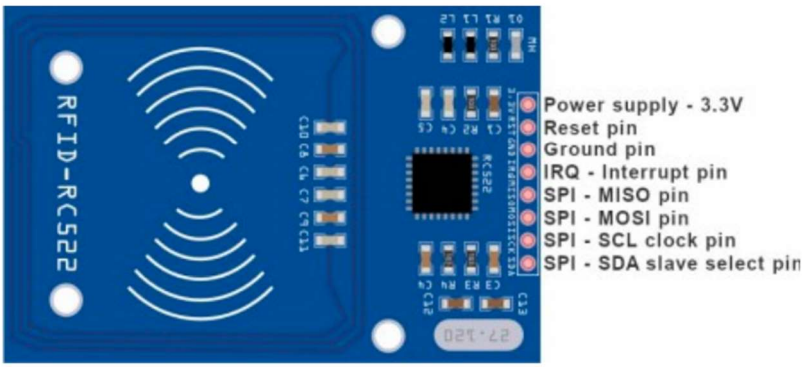
[1]Voor het herschrijven en verduidelijken van teksten werd gebruikgemaakt van ChatGPT (OpenAI, GPT-4), ter ondersteuning van de inhoudelijke en taalkundige uitwerking.

1 Hardwarecomponenten

De hardwarecomponenten vormen de basis van het project. Ze zijn verantwoordelijk voor de registratie van studenten en het efficiënt verwerken van aanwezigheden. In dit hoofdstuk worden de belangrijkste componenten besproken die gekozen worden voor dit project, inclusief hun functies en voordelen. De gekozen hardware moet vooral betrouwbaar zijn, maar ook gemakkelijk te integreren in het systeem.

1.1 RFID-lezer

In dit project wordt een RFID-lezer (Figuur 1) gebruikt, zodat iedere student zichzelf kan registreren in een bepaald lokaal. Dit kan met de eigen studentenkaart of een tijdelijke badge die ook zal werken indien correct gekoppeld.



Figuur 1: RFID-RC522 pinout [1]

RFID staat voor “Radio-frequency identification”. De RFID-lezer is een sensor die gebruikmaakt van elektromagnetische velden om gegevens over korte afstanden over te dragen. Met behulp van RFID-tags kunnen verschillende personen zichzelf identificeren dankzij de unieke code die eraan gekoppeld is. Studentenkaarten bevatten ook een RFID-tag, waardoor deze eveneens kunnen worden gebruikt voor identificatie.

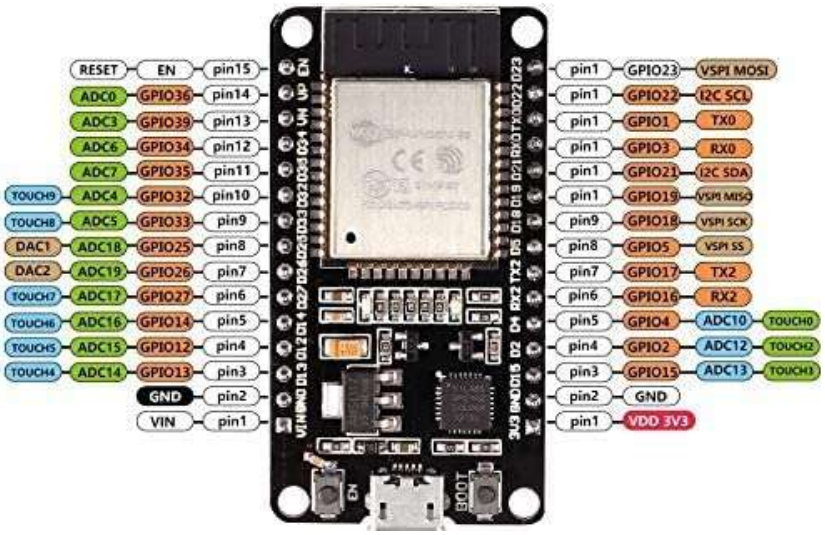
De RFID-lezer bestaat uit twee delen: een microchip die informatie opslaat en verwerkt, en een antenne die de data kan ontvangen en uitzenden naar de tags.

De lezer maakt gebruik van het SPI-protocol (Serial Peripheral Interface) om gegevens uit te wisselen met een microcontroller, zoals een Arduino of een ESP. SPI is een seriële communicatiemethode waarbij data snel en efficiënt over korte afstand kan worden verzonden. De verbinding heeft vier signaallijnen: MISO (Master In Slave Out), MOSI (Master Out Slave In), SCK (Serial Clock) en SS of SDA (Slave Select). De microcontroller werkt hierbij als master en de RFID-lezer als slave.

Om de communicatie goed te laten verlopen, is het van belang dat de klokfrequentie van SPI correct is ingesteld (vaak 4 MHz). Voor de aansturing wordt meestal gebruikgemaakt van een externe bibliotheek, zoals de MFRC522-library die gemakkelijk te installeren is binnen de Arduino IDE. [1]

1.2 ESP32

De ESP32 (Figuur 2) vormt het centrale besturingssysteem van de automaat en zorgt voor de opname, verwerking en doorgifte van informatie. Deze microcontroller wordt specifiek geselecteerd vanwege de krachtige chip in combinatie met de geïntegreerde wifi- en bluetoothfunctionaliteiten.



Figuur 2: ESP32 pinout [3]

Een belangrijk voordeel van de ESP32 ten opzichte van zijn voorganger, de ESP8266, is de standaardondersteuning voor HTTPS-verzoeken. Bij de ESP8266 vereist het opzetten van veilige HTTPS-verbindingen vaak omslachtige oplossingen, zoals het handmatig toevoegen van root-certificaten of het gebruik van aangepaste libraries. Dit kan leiden tot instabiliteit en compatibiliteitsproblemen. De ESP32 daarentegen ondersteunt deze beveiligde communicatie native en betrouwbaar, wat een doorslaggevende factor was bij de hardware keuze. Daarom wordt zowel de automaat als de scanner in dit project gebaseerd op de ESP32.

Daarnaast beschikt de ESP32 over meer datapinnen dan de ESP8266, wat extra aansluitmogelijkheden biedt en dus bijdraagt aan de keuze voor dit model. In de automaat wordt de ESP32 ingezet voor de aansturing van de RFID-lezer, het display, de buzzer, de RGB-led en het mechanisme voor het vrijgeven van badges. Ook verzorgt de microcontroller de communicatie met de backend.

De ESP32 maakt gebruik van de ESP32 WROOM 32-chip, die is uitgerust met een spanningsregelaar en een USB-programmeercircuit. Voor de programmering wordt de

Arduino IDE gebruikt, mede vanwege de actieve gebruikersgemeenschap, brede platformondersteuning en de gebruiksvriendelijkheid. De in- en uitgangspinnen werken op 3,3 V, wat geschikt is voor de toepassingen in dit project. Daarnaast biedt deze microcontroller een wifi van 2,4 GHz met snelheden tot 150 MB/s, BLE (Bluetooth Low Energy), klassieke bluetooth, 34 I/O-pinnen, evenals ondersteuning voor I2C, I2S, ADC, DAC, SPI, UART en PWM. [2]

1.3 Wifirouter

Vermits de ESP's in verbinding moeten staan met de backend, moeten ze aangesloten zijn op een Wi-Fi-netwerk. Hiervoor wordt gebruik gemaakt van een eenvoudige Linksys WRT150N wireless-n home router die fungeert als draadloos toegangspunt binnen het lokale netwerk. De router wordt geconfigureerd met WPA2-PSK (Wi-Fi Protected Access II - Pre-Shared Key) als beveiligingsprotocol, waarbij een sterk wachtwoord wordt gebruikt om ongeautoriseerde toegang te voorkomen.

Om netwerkcommunicatie te vereenvoudigen en afhankelijkheden van DHCP te vermijden, wordt aan elke ESP een statisch IP-adres toegewezen. Deze vaste IP-configuratie werd handmatig ingesteld met vermelding van het juiste subnetmask, gateway en DNS-server. Hierdoor kunnen de apparaten consistent benaderd worden door de backend.

1.4 Display

Er wordt een 2,8 inch TFT-kleurenschermmodule (Figuur 3) gekozen als display aangezien die over een kwalitatieve touchscreen beschikt alsook een lage kostprijs heeft. Het display wordt gebruikt om studenten en docenten de mogelijkheid te bieden om een RFID-tag aan een e-mailadres te koppelen.

Figuur 3: 2,4 inch TFT-kleurenschermmodule pinout [5]

De schermindeling bestaat uit twee secties. Bovenaan en in het midden wordt een lijst weergegeven met e-mailadressen van accounts waaraan nog geen RFID-tag is gekoppeld. Onderaan bevinden zich knoppen om doorheen de e-mailadressen te bladeren en om terug naar home te gaan. Helemaal onderaan bevindt zich een grote knop met de tekst "Koppelen". Eens hierop gedrukt wordt, gaat het systeem na of er een e-mailadres geselecteerd is. Als dit het geval is wordt de tag gekoppeld aan het account en anders wordt de gepaste foutmelding weergegeven.

Op deze manier kan een student of docent na het scannen van een badge het juiste e-mailadres selecteren en de koppeling bevestigen via de knop. De achterliggende software verwerkt deze actie door de RFID-tag in de database aan de overeenkomstige gebruiker te koppelen.

De 2,8-inch TFT-kleurenschermmodule is een component die verschillende figuren kan afbeelden. Het reikt van tekst tot afbeeldingen en allerhande verschillende vormen. Het displaypaneel bevat een dunne-filmtransistor (TFT)-array die de kleur en helderheid van elke pixel regelt. Elke pixel bestaat uit drie subpixels (rood, groen en blauw) die samen een full-color beeld vormen. De achtergrondverlichting, bestaande uit leds langs de randen van het scherm, zorgt voor goede zichtbaarheid onder verschillende lichtomstandigheden.

De ESP-32 stuurt gegevens naar het stuurcircuit, dat deze omzet in signalen voor het displaypaneel. Het drivercircuit beheert de vernieuwingsfrequentie en timing, zodat het beeld of de tekst correct wordt weergegeven.

De aanraakfunctionaliteit wordt mogelijk gemaakt door de resistieve aanraakcontroller XPT2046. Dit systeem detecteert aanraking op basis van druk. Wanneer een gebruiker op het scherm drukt, verandert de elektrische weerstand, waarna de controller dit omzet in digitale coördinaten.

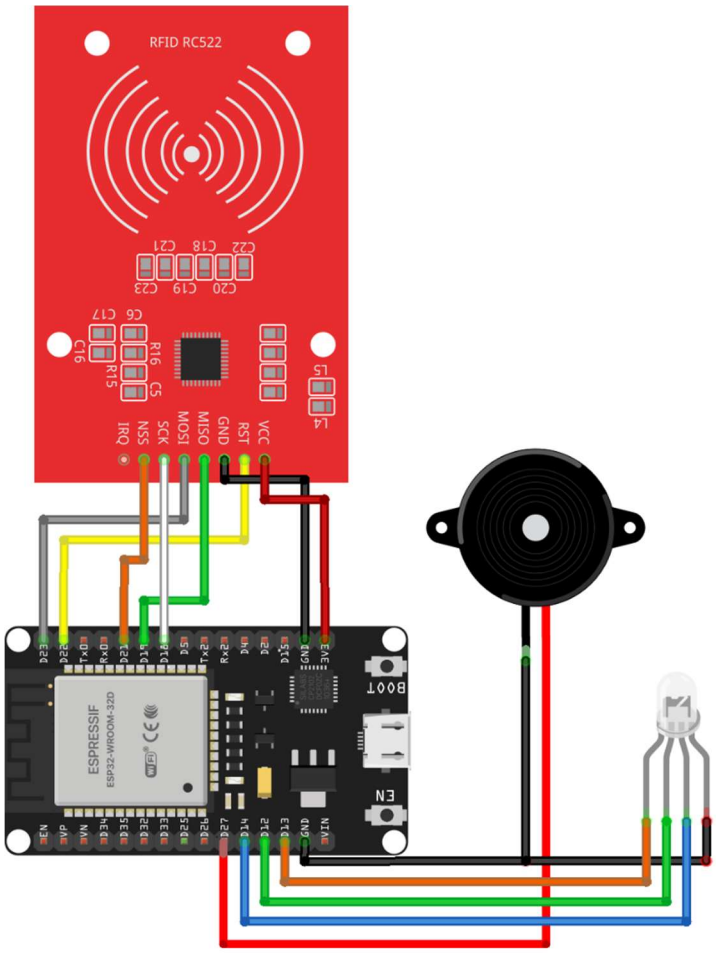
Ook heeft het display de mogelijkheid om gebruik te maken van een SD-kaart. Hiermee kunnen er afbeeldingen of vormen altijd beschikbaar gesteld worden. Echter is dit niet actief in gebruik in dit project. [5]

2 Hardware schema's

In dit hoofdstuk wordt er uitgelegd hoe de hardware geschakeld is en wordt er voor visuele ondersteuning schema's weergegeven. De schema's worden opgesteld in de software Fritzing, een opensource-ontwerptool waarmee elektronische schakelingen visueel kunnen worden weergegeven en gedocumenteerd

2.1 Elektronisch schema scanner

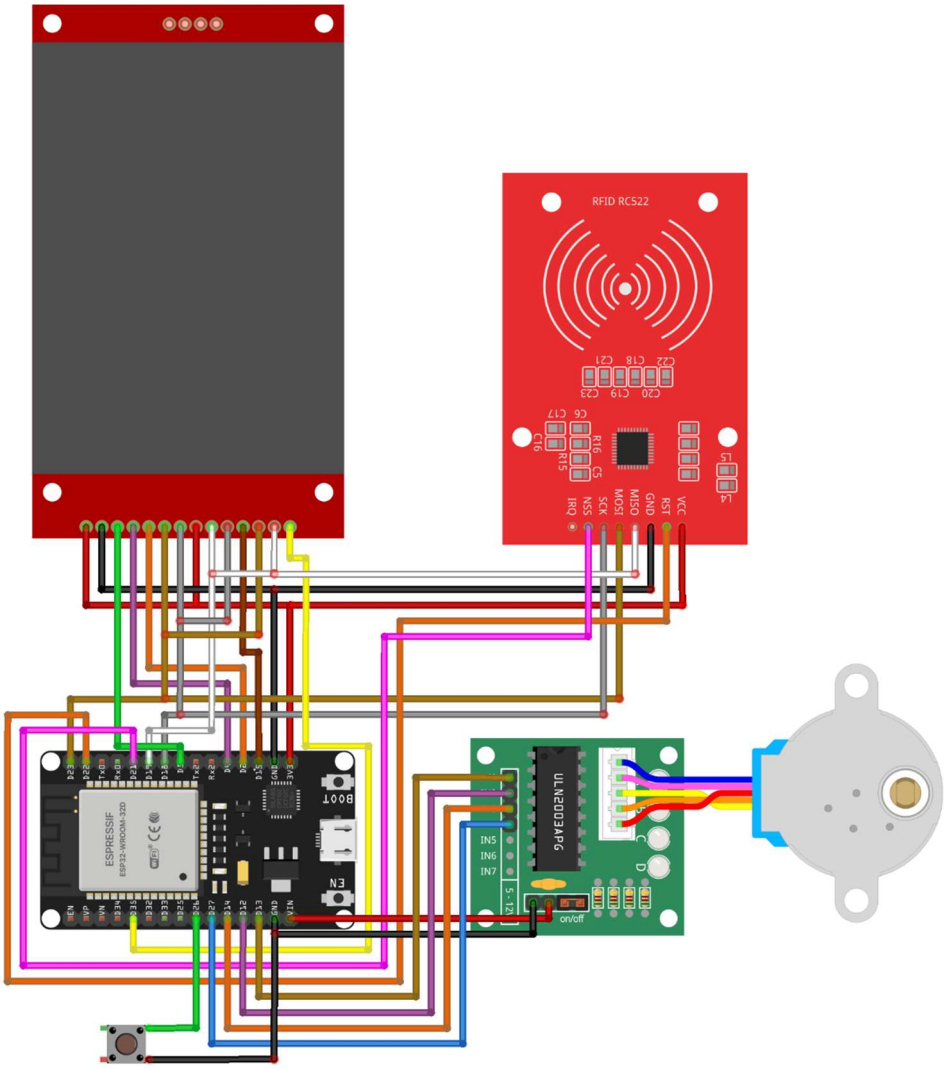
Het schema van de scanner wordt een ESP32-microcontroller verbonden met een RFID RC522-lezer, de communicatie verloopt via het SPI-protocol. Evenals is er een passieve buzzer en een RGB-led voor visuele en auditieve bevestiging voorzien. De lezer heeft vijf datapinnen die worden gebruikt voor de communicatie met de ESP, een GND-pin en een 3.3 V voedingspin. De led heeft drie datapinnen, één voor rood, één voor groen en één voor blauw. Net zoals bij de buzzer is er ook een pin dat wordt verbonden met de GND. De buzzer heeft naast een ground ook een datapin, waardoor hij gevoed wordt als de softwarecode het toelaat.



Figuur 4: Fritzing schema scanner

2.2 Elektronisch schema automaat

Net zoals bij de scanner wordt er een ESP32 gebruikt waar een RFID-lezer aan verbonden is. In het schema van de automaat (Figuur 5) worden dezelfde datapinnen gebruikt voor de lezer als in het schema van de scanner. Naast de lezer wordt er ook een 2,4-inch TFT-display met touchfuntionaliteit verwerkt. Het display heeft elf datapinnen, twee 3.3 V voedingspinnen en één groundpin. De datapinnen omvatten onder andere signaallijnen die dienen voor een SPI-protocol communicatie, alsook pinnen voor de touchfunctionaliteit. Ten slotte wordt er een stappenmotor en een drukknop geïntegreerd in het schema. De stappenmotor drijft de spiraalmechaniek aan, waardoor deze in wijzerzin en tegenwijzerzin kan roteren. De motor wordt aangestuurd door een bijhorende drivermodule die met vier datapinnen verbonden wordt met de ESP32. De module wordt gevoed met de 'V_{IN}' van de microcontroller voor een maximale spanning (3.3 – 5 V) en wordt ook verbonden met de ground. De werking van de stappenmotor wordt handmatig gestart met een drukknop die verbonden wordt op datapin D26.



Figuur 5: Fritzing schema automaat

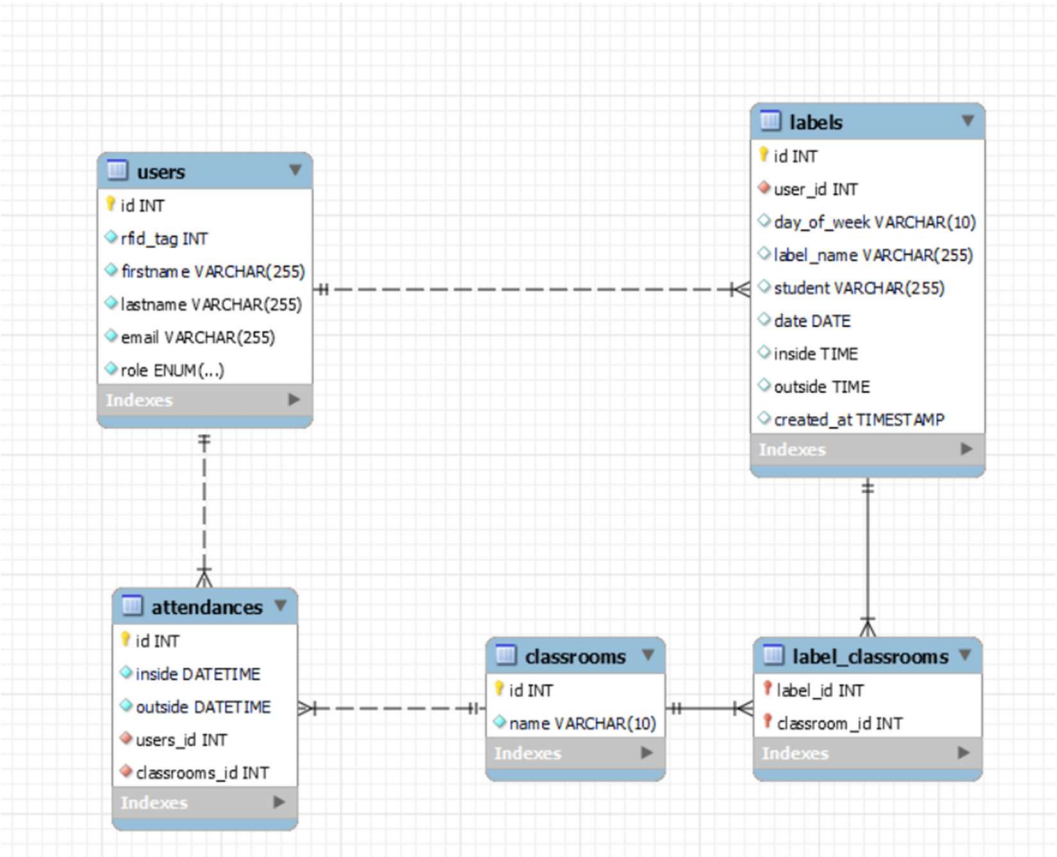
3 Software

Software vormt de kern van het systeem. De toepassing wordt ontworpen om gebruikersinteracties en aanwezigheidsregistraties efficiënt te beheren via een combinatie van webapplicaties en backendprocessen. De software zorgt dat er een stabiele communicatie tussen gebruiker, database en fysieke hardware wordt opgebouwd om zo een veilige en betrouwbare werking te garanderen.

3.1 Database

3.1.1 Overzicht

De database vormt het centrale opslagpunt voor alle gegevens die betrekking hebben op gebruikers, aanwezigheden en systeemkoppelingen. Ze bewaart essentiële informatie over gebruikersprofielen, aanmeldgegevens en de aanwezigheid. Het systeem registreert aankomst en vertrek van gebruikers in het lokaal en stelt docenten in staat om aanwezigheden in een lokaal te controleren. De database slaat de gegevens op in vijf tabellen (Figuur 6).



Figuur 6: Database schema

3.1.2 Structuur

De database bestaat uit verschillende tabellen die gezamenlijk het bijhouden van gebruikersaanwezigheden in klaslokalen mogelijk maken.

De Users-tabel bevat de gegevens van alle gebruikers, studenten, docenten en admins. Per gebruiker wordt een uniek identificatienummer, de gebruikersnaam (volgens het patroon voornaam.naam), het e-mailadres en de bijhorende RFID-tag opgeslagen. Daarnaast wordt er hier een TOTP-secret (Time-based One-Time Password secret) bijgehouden. Doordat er met dit soort authenticatie wordt gewerkt is een aparte logintabel overbodig geworden.

De classrooms-tabel bevat informatie over de fysieke klaslokalen, met een uniek ID en een naam voor elk lokaal. De aanwezigheid van gebruikers in deze lokalen wordt geregistreerd via de Attendances-tabel, die fungeert als een koppeling tussen Users en Classrooms tabellen en registreert de aanwezigheid van gebruikers in lokalen. Voor elke registratie wordt een gebruikers-ID, een lokaal-ID en tijdstempels van aankomst en vertrek opgeslagen.

Deze gegevens worden onder andere gebruikt om aanwezigheden via de website te kunnen raadplegen. Aan deze structuur wordt ook een Labels-tabel toegevoegd. Deze tabel maakt het mogelijk om per gebruiker filters op te slaan, zoals de gewenste datum, aankomst- en vertrektijd, of de dag van de week waarop de gebruiker wil filteren, samen met de naam van de student en het klaslokaal. Hierdoor ontstaat een één-op-veelrelatie tussen users en labels, waarbij één gebruiker meerdere labels kan hebben. Omdat een label bovendien aan meerdere klaslokalen kan worden gekoppeld en een klaslokaal ook in meerdere labels kan voorkomen, is er sprake van een veel-op-veelrelatie tussen Labels en Classrooms. Om deze relatie correct te beheren, werd een extra koppeltabel Label_classrooms voorzien, die beide entiteiten via hun respectieve ID’s met elkaar verbindt.

3.2 Backend

De backend verwerkt alle logica, voert validaties uit en staat in voor gegevensmanipulatie. Deze werkt dus als centrale broker tussen de frontend, de database en de hardwarecomponenten. Zowel de frontend als de microcontrollers communiceren rechtstreeks met de backend via gestructureerde HTTP-verzoeken. Deze verzoeken omvatten onder andere het registreren en valideren van gebruikers, het ophalen van aanwezigheidsgegevens, het koppelen van RFID-tags aan bestaande gebruikersaccounts, en het opvragen van e-mailadressen van gebruikers zonder gekoppelde RFID-tag.

3.2.1 Authenticatieproces

Alle communicatie tussen frontend, backend en hardwarecomponenten verloopt via beveiligde verbindingen, waarbij bijzondere aandacht uitgaat naar de authenticatieprocedures. Zoals eerder aangehaald, maakt het systeem gebruik van Time-based One-Time Password (TOTP)-authenticatie. Tijdens het registratieproces wordt van de gebruiker gevraagd een gebruikersnaam, in het formaat voornaam.naam, en een geldig e-mailadres op te geven. Na succesvolle registratie wordt automatisch een TOTP-secret gegenereerd. Dit secret wordt zowel via een QR-code als in tekstuele vorm aangeboden zodat gebruikers het kunnen scannen of handmatig invoeren in een authenticator-app naar keuze, zoals Google Authenticator, Microsoft Authenticator of Duo Mobile. De gekozen applicatie genereert vervolgens elke dertig seconden een nieuwe, unieke zescijferige authenticatiecode.

De werking van het TOTP-mechanisme is gebaseerd op een gedeeld secret en de actuele tijd. Zowel de server als de authenticator-app beschikken over hetzelfde TOTP-secret. Door dit secret te combineren met de tijd, afgerond op intervallen van dertig seconden, genereert een hashfunctie telkens een tijdelijke code. Tijdens het inloggen voert de gebruiker zowel een gebruikersnaam als de actuele gegenereerde code in waarna de backend deze code valideert door eenzelfde berekening uit te voeren met het opgeslagen secret en de serverklok. Daarbij wordt een beperkte tolerantiewaarde toegepast om kleine afwijkingen in tijdsynchronisatie op te vangen. Als de gegenereerde en ingevoerde codes overeenkomen wordt de gebruiker succesvol ingelogd. Door de beperkte geldigheidsduur van de code wordt de veiligheid aanzienlijk verhoogd en wordt het risico op replay-aanvallen sterk gereduceerd.

Aanvullend op deze TOTP-procedure wordt na registratie automatisch een verificatiemail verstuurd naar het opgegeven e-mailadres. Hiervoor wordt gebruikgemaakt van een speciaal aangemaakte Gmail-account. Pas nadat de gebruiker deze verificatiemail heeft bevestigd, wordt het account geactiveerd en wordt inloggen mogelijk gemaakt. Deze extra verificatiestap vormt een bijkomende beveiligingslaag tegen misbruik en frauduleuze registratiepogingen. [6]

3.2.2 Communicatie met frontend

Naast het authenticatieproces biedt de backend extra functies voor ingelogde gebruikers. De frontend communiceert met de backend door middel van API-aanroepen voor het ophalen van gebruikersgegevens en het uitvoeren van rolgebaseerde acties. Ingelogde gebruikers kunnen hun persoonlijke aanwezigheidsgegevens raadplegen. Afhankelijk van de toegekende gebruikersrol worden hierbij bijkomende gegevens beschikbaar gesteld. Studenten krijgen een overzicht van hun eigen aanwezigheden, docenten hebben inzicht

in de aanwezigheid van studenten die tegelijk met hen aanwezig waren in dezelfde lokalen en administrators kunnen globale aanwezigheidsdata raadplegen van alle gebruikers.

Standaard wordt de rol van student toegekend bij het aanmaken van een nieuw profiel. Als administrator wordt het wel mogelijk om gebruikersrollen te beheren. Hiervoor wordt via de frontend een verzoek naar de backend gestuurd. Enkel geauthentiseerde gebruikers met de adminrol zijn gemachtigd om gebruikersrollen op te vragen en te wijzigen. Alleen vooraf bepaalde rollen (admin, docent, student) zijn toegestaan om de integriteit van het systeem te waarborgen.

Daarnaast biedt het systeem de mogelijkheid aan gebruikers om hun gekoppelde RFID-tag te verwijderen. Dit kan noodzakelijk zijn bij verlies of beschadiging van de tag of wanneer gebruikers hun studentenkaart wensen te gebruiken in plaats van een tijdelijk uitgeleende RFID-tag. De backend verwerkt dit verzoek door de bestaande RFID-associatie in de database te wissen. Na het ontkoppelen kan een nieuwe RFID-tag worden gekoppeld aan hetzelfde account, zonder dat herregistratie vereist is. Deze flexibiliteit verhoogt de gebruiksvriendelijkheid en vermindert administratieve tussenkomst.

Alle gegevensuitwisseling verloopt via beveiligde sessies. Verzoeken zonder geldige sessie worden geweigerd, wat het risico op ongeoorloofde toegang aanzienlijk beperkt. De backend controleert bij elk verzoek op de aanwezigheid en geldigheid van de sessievariabelen zoals e-mailadres en gebruikersrol. Indien de vereiste sessiecontext ontbreekt, wordt een foutmelding gegenereerd en de toegang geblokkeerd.

3.2.3 Filteren en sorteren

Wanneer een gebruiker via de frontend filters toepast of een sorteerparameter kiest om aanwezigheidsgegevens op te vragen, verzendt de browser een HTTP GET-verzoek naar het endpoint /data. De filters en sorteervoorkeuren worden meegegeven als queryparameters in de URL. Dit kunnen bijvoorbeeld velden zijn als ?student=Jan+Janssens&date=2024-12-01&sortBy=inside&sortDir=desc&page=2&limit=20. De backend ontvangt deze informatie als onderdeel van $_GET.

In de backend start een functie met het opstarten van een sessie en het verifiëren van de rol en identiteit van de gebruiker. Afhankelijk van de gebruikersrol (student, docent, admin) wordt een verschillende SQL-query gegenereerd. Bij een student worden enkel diens eigen aanwezigheidsgegevens opgehaald. Bij een docent zoekt het systeem naar de gebruikers die gelijktijdig aanwezig waren in hetzelfde lokaal. Voor een admin kunnen de gegevens van alle gebruikers gefilterd worden op basis van de meegegeven parameters.

De backend combineert filters uit de URL met eventuele eerder opgeslagen filterlabels (bijvoorbeeld via label_id) die gebruikers kunnen beheren. Hierdoor kunnen filters zoals dag van de week, leslokaal-ID's, studentennaam, datum of tijdstippen zoals inside=08:00

of outside=12:00 samen worden toegepast. Het systeem verwerkt die filters in een SQL WHERE-clausule die dynamisch opgebouwd wordt in een methode.

Voor sortering gebruikt de backend parameters zoals sortBy en sortDir. De toegelaten sorteerwaarden zijn beperkt tot kolommen zoals inside, classroom_name, en student_name. Deze parameters worden via een variabele in de SQL-query gezet. Er wordt echter gecontroleerd of de gevraagde kolom wel degelijk geldig is om SQL-injectie te vermijden.

De backend past ook paginatie toe via LIMIT en OFFSET, berekend op basis van de pagina en limietwaarden. Dit betekent dat enkel de gevraagde subset aan gegevens naar de frontend gestuurd wordt, bijvoorbeeld enkel 20 records voor pagina twee. Tegelijk wordt via een tweede query (COUNT(*)) het totaal aantal resultaten berekend, zodat de frontend het aantal beschikbare pagina's kan inschatten.

Het filteren en sorteren in de backend is om meerdere redenen voordeliger dan dat over te laten aan de frontend. Ten eerste wordt de backend rechtstreeks verbonden met de databank en kan dus veel efficiënter filteren op grote datasets. Indien filtering pas na het laden in de frontend gebeurt, zou de server telkens alle gegevens moeten verzenden, wat bandbreedte verspilt en traagheid veroorzaakt, vooral op mobiele netwerken. Bovendien waarborgt filtering op de backend dat alleen toegelaten gegevens getoond worden, bijvoorbeeld beperkt tot de eigen studenten of enkel het afgelopen jaar, wat essentieel is voor privacy en beveiliging.

De filteropbouw via URL laat een intuïtieve debugging toe, want een query als
/data/?student=Janssens&inside=08:00&weekday=Monday
maakt het mogelijk om onmiddellijk te zien waarop er gefilterd wordt. Tegelijk kunnen filters gecombineerd worden met opgeslagen labels die gebruikers eerder bewaarden. Die labels worden intern herleid tot specifieke filtervelden en via joins met andere tabellen zoals Label_classrooms geïntegreerd in de query.

Zo ontstaat een schaalbaar, flexibel en veilig systeem waarbij gebruikers hun eigen weergavecriteria bepalen zonder de backend te belasten met irrelevante data of de frontend te overladen met onnodige rekenkracht.

3.2.4 Communicatie met hardware

Voor een veilige communicatie tussen de ESP-hardware en de backend maakt elke ESP bij een verzoek gebruik van een HMAC-token, bestaande uit een geheime sleutel en een timestamp. Dit voorkomt dat onbevoegde clients zomaar toegang kunnen krijgen tot de ESP-endpoints. De backend valideert deze HMAC-header en controleert of de timestamp recent is, zodat alleen geldige en geautoriseerde verzoeken worden verwerkt.[8]

De communicatie tussen de backend en de RFID-hardware wordt afgehandeld door middel van API-aanroepen die de gebruiker toestaan om een RFID-tag te koppelen aan een account en vervolgens de tag te scannen bij binnenkomst en vertrek uit een lokaal.

Dit wordt allemaal afgehandeld door de ESP's. Die sturen een GET-verzoek naar de backend, waarna de backend alle geregistreerde e-mailadressen zonder gekoppelde RFID-tag als antwoord terugstuurt. Wanneer de gebruikers aan het display hun tag scannen en vervolgens op het touchscreen hun e-mailadres selecteren, stuurt de ESP een POST-verzoek met de RFID-tag en het bijbehorende e-mailadres. Bij een succesvolle koppeling wordt de database bijgewerkt en wordt de RFID-tag aan de gebruikersgegevens toegevoegd. Als de gebruiker niet wordt gevonden of de RFID-tag al is gekoppeld, wordt er een foutmelding geretourneerd.

Zodra de RFID-tag is gekoppeld kunnen de gebruikers hun tag gebruiken om aanwezigheidsregistraties te verrichten. Wanneer gebruikers hun RFID-tag scannen bij binnenkomst of vertrek uit een lokaal, wordt de tag samen met de betreffende lokaalinformatie naar de backend gestuurd. De backend verifieert of de ontvangen gegevens geldig zijn en verwerkt de aanwezigheid zodat deze correct in de database terechtkomen.

Bij beide processen, zowel het koppelen van de RFID-tag als het scannen van de tag, worden foutmeldingen afgehandeld. Fouten in de communicatie met de server, zoals ontbrekende gegevens of serverproblemen, resulteren in een passende HTTP-statuscode en een foutbericht.

Deze aanpak garandeert dat de integriteit van de communicatie gewaarborgd blijft, doordat elke stap van de interactie met de hardware grondig wordt gecontroleerd op geldigheid en autorisatie.

3.3 Frontend

De frontend van de toepassing vormt de gebruikersinterface en bouwt de interactie met de backend-API op via asynchrone HTTPS-verzoeken. Het geheel wordt opgebouwd met TypeScript en werkt browsergebaseerd. De structuur volgt een modulaire aanpak waarbij verantwoordelijkheden worden gescheiden over verschillende componenten en functies. Interactie met gebruikersdata gebeurt aan de hand van formulieren, filters en dynamische weergegeven content op de webpagina.

3.3.1 Header

De header bevat een navigatiebalk die zich aanpast op basis van de aanmeldstatus en rol van de gebruiker. Met behulp van een hamburgericoon wordt de navigatie responsive, wat de bruikbaarheid op mobiele apparaten verhoogt.

Bepaalde navigatie-items, zoals toegang tot administratieve functionaliteiten, worden enkel zichtbaar indien de gebruiker de rol admin bezit. Deze toegangscontrole wordt deels client-side afgedwongen via gegevens opgeslagen in sessionStorage, maar wordt eveneens server-side gevalideerd via sessiegegevens om ongeautoriseerde toegang te verhinderen.

De welkomstboodschap wordt dynamisch weergegeven via de klasse welcome-message (Codefragment 1Codefragment 1). Bij het inloggen worden de gebruikersgegevens automatisch opgehaald en verwerkt in de navigatie.

```
1 const welcomeMessages = document.querySelectorAll(".welcome-message")
2 welcomeMessages.forEach((el) => (el.textContent = firstName!));
```

Codefragment 1: Klasse welcome-message

3.3.2 Gegevensuitwisseling met de backend

Gegevenscommunicatie gebeurt via de fetch-API met expliciete configuratie van methoden, headers en eventueel bodycontent in JSON-formaat. De meeste interacties vereisen authenticatie, wat afgedwongen wordt door het gebruik van credentials: "include", waardoor sessiecookies worden meegestuurd. De API-url wordt dynamisch opgehaald uit omgevingsvariabelen via import.meta.env.

Een typisch voorbeeld van zo'n verzoek wordt aangetroffen in het loginproces. Daar wordt een JSON-body verstuurd met inloggegevens. Na validatie op de backend worden gegevens zoals rol en gebruikersnaam lokaal opgeslagen in sessionStorage, wat persistentie biedt binnen de actieve sessie. Daarna volgt een redirect naar de gebruikersinterface.

3.3.3 Sessiebeheer en toegangscontrole

Sessiebeheer gebeurt zowel aan de client- als aan de serverzijde. Na een succesvolle login worden gebruikersgegevens zoals voornaam, e-mailadres en rol opgeslagen in de sessionStorage van de browser. Deze gegevens worden gebruikt om de gebruikersinterface dynamisch aan te passen. Zo kunnen bepaalde elementen worden getoond of verborgen op basis van de rol van de gebruiker. Een gebruiker met een admin of docent rol krijgt bijvoorbeeld toegang tot een extra beheerderstabblad. Tegelijkertijd wordt er op de backend een serversessie gestart waarin onder andere het e-mailadres en de gebruikersrol worden opgeslagen. Deze serversessie wordt gebruikt om toegang tot API-routes te controleren, zodat gebruikers zonder geldige sessie geen toegang krijgen tot beschermde gegevens.

3.3.4 Validatie en gebruikersinteractie

Bij alle formulieren worden de invoervelden eerst client-side gevalideerd voordat een verzoek wordt verstuurd. De validatie controleert onder andere op de aanwezigheid van alle verplichte velden. Voor de registratie- en inlogformulieren wordt daarnaast specifiek gecontroleerd op de structuur van het e-mailadres en het verwachte formaat van de gebruikersnaam.

Na het succesvol verzenden van een formulier verschijnt er een toastmelding om de gebruiker hiervan op de hoogte te brengen, wat bijdraagt aan een positieve gebruikerservaring. [9]

Bij het inloggen gebeurt dit niet, omdat de gebruiker in dat geval direct wordt doorgestuurd naar de startpagina. Bij registratie wordt er geen redirect uitgevoerd. In plaats daarvan wordt een QR-code weergegeven op de registratiepagina. Deze QR-code is bedoeld voor TOTP-authenticatie en wordt gepresenteerd als een afbeelding, vergezeld van een handmatige sleutel die de gebruiker kan kopiëren naar het klembord. Dit kopiëren gebeurt via navigator.clipboard.writeText, een browser-API die asynchroon werkt en foutgevoelig kan zijn.

3.3.5 Contactpagina

De contactpagina bevat een formulier waarmee gebruikers eenvoudig een bericht kunnen versturen. Dit formulier wordt verbonden met Formspree, een externe dienst die het versturen van formulierdata per e-mail faciliteert zonder dat een eigen backend voorzien hoeft te worden (Codefragment 2**Fout! Verwijzingsbron niet gevonden.**).

```
1 <form action="https://formspree.io/f/xqapjwqw" mehtod="POST">
```

Codefragment 2: Formspree POST form

In tegenstelling tot andere HTML-formulieren, wordt bij dit formulier de volledige afhandeling geregeld via de HTML-tag zelf. De gegevens worden automatisch via een expliciete POST-request verzonden naar de opgegeven action-URL van Formspree, zonder dat daar extra TypeScript of backendcode voor nodig is.

Wanneer een gebruiker is ingelogd worden de naam en e-mail van de gebruiker automatisch ingevuld in het formulier op de contactpagina, waardoor gebruikers tijd besparen en de kans op fouten kleiner wordt. [7]

3.4 Microcontrollersoftware

3.4.1 Scanner ESP32

De scanner wordt volledig geprogrammeerd binnen de Arduino IDE en maakt gebruik van de ESP32-microcontroller. De code bevat verschillende functies die samenwerken om de aanwezigheid van gebruikers via RFID-technologie te registreren en te versturen naar de backend. De code wordt opgebouwd uit een aantal gestructureerde onderdelen die samen zorgen voor correcte initialisatie, netwerkcommunicatie, uitlezen van RFID-tags en het verzenden van data in JSON-formaat via HTTP POST-verzoeken.

Allereerst worden de noodzakelijke bibliotheken toegevoegd, waaronder SPI en MFRC522 voor het aansturen van de RFID-lezer, WiFi en HTTPClient voor de Wifi-verbinding en de HTTP-communicatie en tot slot nog ArduinoJson voor het structureren van gegevens in JSON-formaat. Vervolgens worden er enkele constante waarden en variabelen gedeclareerd.

Een belangrijke functie bij het opstarten is setup_wifi() (Codefragment 3Codefragment 3Codefragment 3). Deze functie zorgt voor het verbinden met het opgegeven wifinetwerk. Als de verbinding succesvol is, wordt het lokale IP-adres weergegeven. De verbinding heeft een maximale wachttijd van 15 seconden, waarna de status wordt geëvalueerd.

```
void setup_wifi() {
  Serial.begin(115200);
  Serial.print("Verbinden met WiFi: ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  int timeout = 30;
  while (WiFi.status() != WL_CONNECTED && timeout > 0) {
    delay(500);
    Serial.print(".");
    timeout--;
  }

  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nWiFi verbonden!");
    Serial.print("IP: ");
    Serial.println(WiFi.localIP());
  } else {
    Serial.println("\nWiFi verbinding mislukt!");
  }
  Serial.println();
}
```

Codefragment 3: Void setup_wifi()

De setup()-functie initialiseert de seriële communicatie, stelt de buzzerpin in als uitvoer en start het SPI-protocol op. Dat protocol is essentieel voor communicatie met de RFID-lezer. De RFID-lezer wordt vervolgens geïnitialiseerd via de methode PCD_Init(). In de setup()-functie wordt ook de setup_wifi()-functie aangesproken, om een verbinding op te bouwen tussen de microcontroller en het wifinetwerk.

In de loop()-functie wordt de kernfunctionaliteit uitgevoerd. Hier wordt voortdurend gecontroleerd of er een nieuwe RFID-kaart wordt aangeboden. Als er een kaart wordt gedetecteerd, wordt het unieke ID (UID) van de kaart uitgelezen. Deze wordt opgeslagen

als een string. Als er een verbinding is die eerder in de setup opgesteld werd, dan worden de gegevens naar de server gestuurd. Indien dit niet het geval is, wordt het verzenden van data overgeslagen. Als de verbinding wel actief is, wordt er een WiFiClient en HTTPClient instantie aangemaakt. De URL voor de POST-aanvraag wordt samengesteld op basis van het IP-adres, de poort en het opgegeven serverpad.

Als de student gescand heeft wordt er een JSON-object opgebouwd met daarin de UID van de RFID-kaart en het lokaal waarin de scanner zich bevindt. Dit JSON-object wordt geserialiseerd tot een string. Het verzenden van de gegevens gebeurt via een HTTPS-POST-verzoek, waarbij de gestructureerde JSON als payload wordt meegestuurd.

De HTTP-header wordt ingesteld op application/json, zodat de server de inhoud correct interpreteert. Na het verzenden wordt de HTTP-responscode gecontroleerd en gelogd. Bij een succesvolle transmissie wordt een toon geproduceerd via de buzzer als auditieve bevestiging. Als de POST-aanvraag mislukt, wordt dit eveneens weergegeven met een foutmelding in de seriële monitor.

Tot slot wordt de RFID-kaartsoftware correct afgesloten met de functie PICC_HaltA(), waarna een korte vertraging volgt voor de volgende iteratie van de lus. Dit proces herhaalt zich continu zolang het apparaat actief is.

3.4.2 Automaat ESP32

Ook voor de automaat wordt een ESP32 gebruikt als microcontroller om deze te laten functioneren. Ook hier wordt er weer gebruik gemaakt van alle functies om een internetverbinding op te bouwen, de RFID-lezer te laten werken en om de nodige POST-verzoeken te doen in JSON-formaat. Echter heeft deze ESP32 nog veel bijkomende functies. Zo bevat deze drie verschillende functies om een bepaald beeldscherm op te maken: een homescherm, een scanscherm en nog een extra functioneel scherm om de opgehaalde e-mails zonder gekoppelde tag op te tonen. Andere functies zijn bijvoorbeeld nog de touchscreen, GET en POST functies om e-mailadressen op te halen en tags te koppelen aan een account.

Net zoals bij de scanner initialiseert de setup()-functie hier ook alle nodige protocollen en functies. Zowel de RFID als het touchscreen van het TFT-display gebruiken beide de SPI-bus. Hierdoor moet er wel geregeld worden wanneer welk component actief mag zijn zodat ze elkaar niet storen of blokkeren. In de setup()-functie wordt de chip select pin (cs pin) van de RFID laag gemaakt en die van de touchscreen hoog, op deze manier wordt de RFID-lezer geactiveerd en staat de touchscreen tijdelijk nog uitgeschakeld. Via de functies enableRFID() en enableTouch() wordt het mogelijk om doorheen de code op een makkelijk manier te wisselen tussen welk component actief is en welke niet.

In de loop()-functie zorgt een timer ervoor dat er continu gewisseld wordt tussen de schermen die op het display zichtbaar zijn. Gedurende 15 seconden wordt het

homescherm getoond. Hierop staat er een QR-code die naar de EduBadge website leidt en de bijhorende “Scan QR-code” tekst. Eens de 15 seconden verlopen zijn wordt het scherm veranderd naar het scanscherm. Dan wordt er 5 seconden lang de boodschap “Scan uw kaart” op het display weergegeven.

Doorheen de loop blijft de RFID-lezer continue actief. Eens er een UID wordt gedetecteerd komt er “Kaart gescand” op het display en wordt deze UID opgeslagen in een bijhorende variabele. De functie enableTouch() wordt uitgevoerd waardoor de RFID wordt uitgeschakeld en de touch wordt ingeschakeld. Vervolgens wordt de functie sendGet() uitgevoerd waarbij alle e-mailadressen worden opgehaald uit de database waaraan nog geen RFID-tag gekoppeld is. Deze worden vervolgens allemaal weergegeven op het scherm met e-mails. Hier moet de gebruiker via de touchscreen het eigen e-mailadres selecteren en onderaan op de knop “selecteren” drukken. Na de bevestiging goedgekeurd te hebben zal de functie sendPost() uitgevoerd worden. Met deze functie wordt de uitgelezen UID samen met het geselecteerde e-mailadres als JSON naar de backend gestuurd zodat de tag gekoppeld wordt. Eens deze POST succesvol uitgevoerd wordt schakelt het display weer over naar het homescherm en wordt de functie enableRFID() aangeroepen zodat de RFID-lezer weer geactiveerd wordt en de loop klaar staat om nieuwe tags uit te lezen en registraties uit te voeren.

4 Securitymaatregelen

De betrouwbaarheid van aanwezigheidsregistratie via EduBadge hangt in sterke mate af van de informatiebeveiliging binnen het gehele systeem. Om ongeautoriseerde toegang, gegevensverlies en manipulatie tegen te gaan worden beveiligingsmaatregelen toegepast op drie essentiële componenten: de website, de database en de communicatie met de microcontrollers. Elke component vormt een potentieel aanvalsoppervlak en wordt daarom individueel en in samenhang beveiligd volgens de gangbare principes.

4.1 Website

De webapplicatie vormt het primaire interactiepunt voor gebruikers zoals studenten, docenten en beheerders. Hoewel de applicatie geen gegevens opslaat, verwerkt ze gevoelige informatie zoals gebruikersidentiteiten en aanwezigheidshistorieken via een backend die met een database wordt verbonden. Toegang tot deze gegevens vereist robuuste authenticatie- en autorisatiemechanismen.

Voor tweefactorauthenticatie wordt gebruikgemaakt van TOTP (Time-based One-Time Password). Tijdens de registratie ontvangt de gebruiker een geheim (TOTP-secret) dat kan worden ingelezen in een authenticator-app zoals Google Authenticator of Microsoft Authenticator. Op basis van dit geheim en de huidige tijd wordt elke 30 seconden een nieuwe zescijferige code gegenereerd. De gebruiker moet deze code, samen met zijn e-mailadres, invoeren om succesvol in te loggen. De backend voert een gelijkaardige berekening uit op basis van de serverklok en het opgeslagen geheim, met tolerantie voor kleine tijdsverschillen. Door dit dynamische mechanisme wordt het risico op afgeluisterde of hergebruikte codes aanzienlijk.

Daarnaast ontvangt de gebruiker na registratie een verificatiemail met een unieke activatielink. Pas na bevestiging van deze link wordt de account geactiveerd en kan de gebruiker aanwezigheid registreren. Dit voorkomt de aanmaak van nepaccounts en verhoogt de betrouwbaarheid van het gebruikersbestand.

De toegangscontrole wordt strikt toegepast op basis van gebruikersrollen. De interface en functionaliteiten passen zich aan op basis van de actieve sessie, waarbij client-side filtering (middels sessionStorage) wordt gecombineerd met server-side validatie. Alleen gebruikers met een geautoriseerde rol krijgen toegang tot kritieke beheermodules. Deze validatie voorkomt privilege-escalatie via manipulatie van clientdata.

Sessiebeheer garandeert dat elk backendverzoek afkomstig is van een actieve, geldige sessie. Verzoeken zonder correcte sessievariabelen worden automatisch geweigerd. Sessies bevatten sleutelgegevens zoals gebruikersrol en e-mailadres en worden beveiligd opgeslagen.

Bij registratie en login wordt de invoer zowel aan client-side als server-side gevalideerd op volledigheid en correcte syntax (zoals het formaat van het e-mailadres), wat foutieve of kwaadaardige invoer vroegtijdig tegenhoudt en serverbelasting verlaagt.

Alle communicatie verloopt via HTTPS, waarmee de gegevens tussen frontend en backend worden versleuteld. Hierdoor wordt afluisteren of manipulatie van dataverkeer door derde partijen voorkomen.

4.2 Database

De database bevat gevoelige persoonsgegevens en aanwezigheidregistraties, die beschermd moeten worden tegen ongeautoriseerde toegang, manipulatie en datalekken. De frontend heeft geen directe toegang tot de database; alle interactie verloopt uitsluitend via de backend, die fungeert als enige toegangslaag tot de databank.

TOTP-secrets worden niet in leesbare vorm opgeslagen, maar gehasht met een cryptografisch veilige functie. Hoewel het hierbij niet om conventionele wachtwoorden gaat, is de vertrouwelijkheid van deze gegevens essentieel. Indien een aanvaller toegang verkrijgt tot een opgeslagen secret, kan deze eenvoudig worden toegevoegd aan een authenticator-app waarmee elke 30 seconden een geldige inlogcode gegenereerd kan worden. Door hashing wordt de impact van een eventueel datalek sterk beperkt aangezien het reconstrueren van het oorspronkelijke geheim nagenoeg onmogelijk wordt zonder de juiste sleutel.

Alle database-interacties maken gebruik van prepared statements met parameterbinding. Deze methode voorkomt SQL-injectieaanvallen door ervoor te zorgen dat gebruikersinvoer nooit direct in de SQL-query wordt geïnjecteerd. In plaats daarvan wordt invoer als parameters doorgegeven aan vooraf gecompileerde statements, wat de uitvoerstructuur van de query ongewijzigd houdt.

De databasegebruiker die door de backend wordt gebruikt, heeft enkel minimale toegangsrechten (Least Privilege Principle). Hierdoor kan de applicatie enkel de noodzakelijke bewerkingen uitvoeren: gegevens opvragen, invoegen, bijwerken en verwijderen. Administratieve acties zoals het wijzigen van tabellen, het toevoegen van gebruikers of het uitvoeren van systeemopdrachten zijn expliciet verboden.

4.3 Communicatie met microcontrollers

De microcontrollers, de ESP32-modules, verzorgen de registratie van RFID-scans bij aanvang van lessen en de koppeling van RFID-tags aan accounts. De communicatie tussen deze modules en de backend introduceert specifieke risico's, zoals spoofing en replay-aanvallen.

Authenticatie van apparaten gebeurt via HMAC-tokens. Bij elk verzoek genereert de ESP een HMAC-header op basis van een gedeeld geheim en een timestamp. De backend valideert deze gegevens op correctheid en recentheid. Verzoeken met een verlopen timestamp of onjuiste HMAC worden verworpen. Deze techniek voorkomt dat een kwaadwillende met onderschepte gegevens een geldig verzoek kan nabootsen.

Iedere ESP beschikt over een statisch IP-adres binnen het netwerk. Deze configuratie vereenvoudigt de beveiliging en monitoring, aangezien verkeer afkomstig van onverwachte adressen als verdacht wordt beschouwd. De netwerktoegang wordt verder beschermd via WPA2-PSK-encryptie, waardoor alleen apparaten met de juiste netwerksleutel kunnen deelnemen aan het verkeer.

Bij elke RFID-scan valideert de backend of het tag-ID geldig is, of de tag gekoppeld is aan een ingeschreven gebruiker en of het gescande lokaal overeenkomt met een geregistreerd lokaal. Hierdoor wordt vermeden dat gescande tags zonder fysieke aanwezigheid of in het verkeerde lokaal worden geregistreerd. Manipulatiepogingen worden hiermee proactief geblokkeerd.

Foutafhandeling wordt expliciet geïmplementeerd. In geval van verbindingsproblemen, ontbrekende gegevens of validatiefouten genereert de server passende HTTP-statuscodes en foutberichten. Deze meldingen kunnen via logica in de ESP worden geïnterpreteerd zodat foutafhandeling lokaal kan plaatsvinden doormiddel van foutmeldingen op het display of via een led en buzzer bij de scanner. [8]

5 Ontwerpen

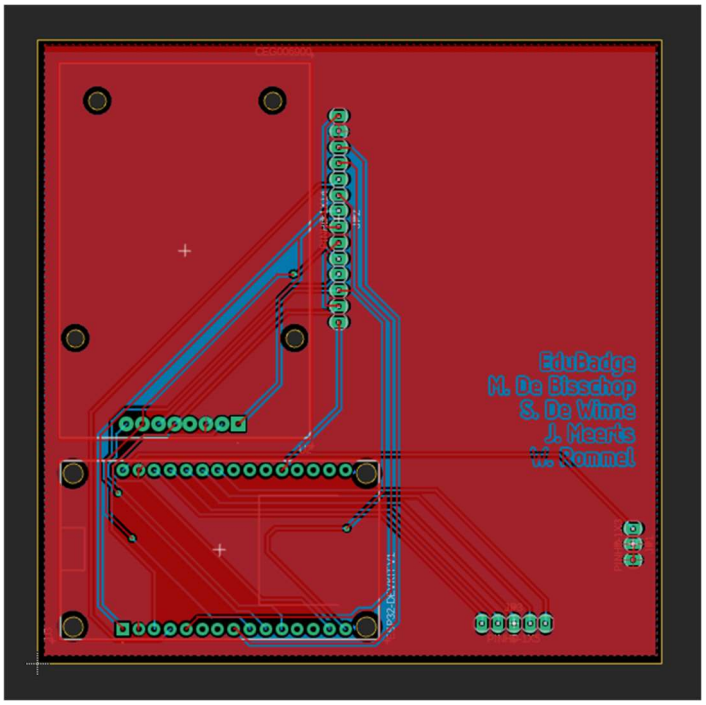
5.1 Printplaat

Er wordt bewust gekozen om printplaten te ontwerpen. Deze zorgen dat het project een stuk compacter wordt dan bij het gebruik van breadboards of testprints. Een extra voordeel is dat de verbindingen veel betrouwbaarder en overzichtelijker zijn.

De componenten zitten dicht bij elkaar om het ontwerp zo compact mogelijk te houden. De dimensies van de printplaten zijn telkens minder dan 10 centimeter. Een extra korting wordt toegepast doordat minder ontwikkelwerk vereist is van de ontwikkelaar.

Er zijn twee verschillende printplaten ontworpen voor het project. Eén voor de scanner die bij de ingang van het lokaal komt en de andere zit in de automaat waarvan er ~1 per gang komt. Beide bevatten plaats voor zowel een ESP32-module als een RFID-lezer. Dit zijn dan ook de basiselementen waar het project ronddraait. Ook wordt er gebruik gemaakt van een GND-vlak. Dit vlak zorgt onder andere voor minder ruis, minder elektromagnetische interferentie en onderdrukking van ongewenste spanningsverschillen tussen GND-punten.

Op de printplaat voor de automaat (Figuur 7) is er plaats voor het touchscreen-display voorzien. Ook is er plaats voor acht aparte headers toegevoegd. Vijf van deze headers zijn

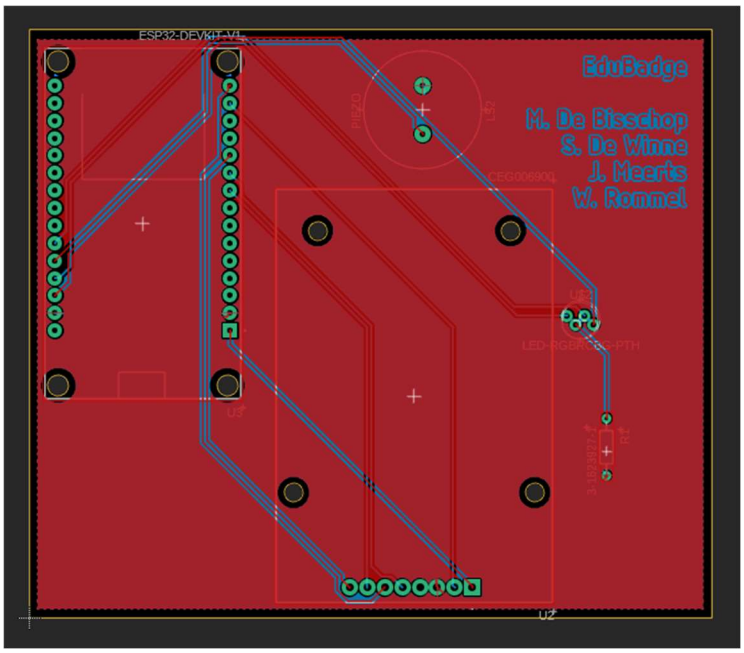


Figuur 7: printplaat automaat

datapinnen voor de stappenmotor en de knop. De overige drie bevatten 1 VCC-punt en 2 GND-punten, deze geven stroom aan de stappenmotor en knop.

De printplaat voor de scanner (

Figuur 8) daarentegen bevat extra plaats voor een buzzer en een RGB-led die verbonden worden met een weerstand. Deze weerstand dient om te zorgen dat er niet te veel stroom door de led stroomt. De ESP32 wordt zo hoog mogelijk geïnstalleerd om de USB-type-C-poort zo makkelijk mogelijk te bereiken met de kabel die deze microcontroller van stroom



voorziet.

Figuur 8: printplaat scanner

5.2 Design

De behuizing van zowel scanner als automaat wordt ontworpen met een 3D-printer. Er wordt gekozen voor een zwart filament omdat hiermee onnauwkeurigheden bij het printen minder zichtbaar zijn.

De scanner (**Fout! Verwijzingsbron niet gevonden.**) wordt zo compact mogelijk gemaakt, zodat ze zo min mogelijk hinder veroorzaakt wanneer ze bij de deur van de lokalen hangt. Eerst wordt de achterkant aan de muur gehangen met behulp van de ontworpen gaten. Vervolgens kan de printplaat en het deksel van het ontwerp vastgemaakt worden. Het deksel wordt vastgemaakt met een kliksysteem dat een makkelijke montage, maar moeilijke demontage toelaat. Op deze manier kan de scanner moeilijk geopend worden door onbevoegden. Er wordt een kleine tool voorzien om het deksel toch op een relatief makkelijke manier te demonteren.

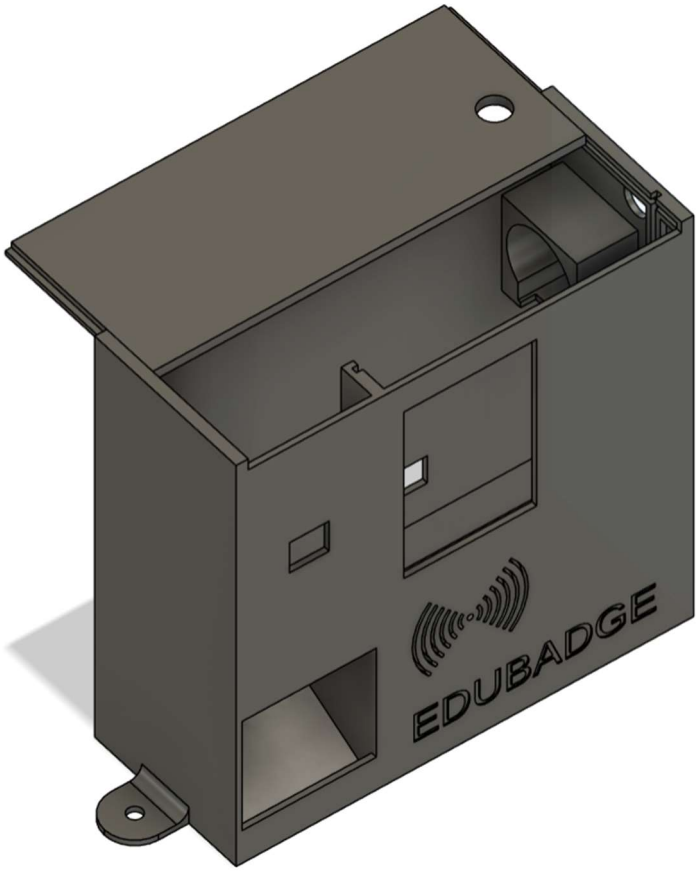
In het deksel staat een RFID-logo zodat duidelijk is waar er moet gescand worden. Er wordt ook een plaats voorzien voor de led, die visueel aangeeft of een RFID-tag al dan niet correct gescand wordt. Aan de achterkant van het ontwerp wordt plaats voorzien voor een voedingskabel.

Figuur 9: Scanner design

De automaat (Figuur 10) werkt met een schuifsysteem waarbij de printplaat langs de bovenzijde in de automaat wordt geschoven. Het display wordt zichtbaar door een gat in de voorkant van de automaat dat speciaal hiervoor ontworpen is. Onder het display bevindt zich het RFID-logo waar er gescand kan worden. Aan de rechterkant bevindt zich een holte waar de stappenmotor in kan gemonteerd worden. Aan deze stappenmotor wordt een metalen spoel gemonteerd waar de extra RFID-tags hangen. Wanneer deze

stappenmotor geactiveerd wordt door de microcontroller, zullen deze tags zich langzamerhand naar de linkerkant van de spoel begeven. Per omwenteling die de spoel maakt, zal er een RFID-tag uit het gat vallen dat zich linksonder de automaat bevindt. Er wordt een smalle rand voorzien om te voorkomen dat de tags volledig uit de automaat vallen, terwijl ze toch eenvoudig uitneembaar blijven. Ook wordt er aan de voorkant een gat voorzien voor de knop die de stappenmotor zal laten draaien. Aan de achterkant bevindt zich een gat dat bedoeld is voor de voedingskabel van de ESP32. Achteraf wordt het deksel langs achter naar voor geschoven om de automaat te sluiten.

Bijkomend wordt de automaat voorzien van twee gaten voor een hangslot, dat de automaat beveiligt. Onderaan zijn er ook twee montagegaten toegevoegd.



Figuur 10: Automaat design

6 Kostprijsberekening

Hoewel er geen vast budget werd opgelegd door de opdrachtgever, geldt de richtlijn om de kosten tot een minimum te beperken. Bij het ontwerp en de implementatie wordt hiermee rekening gehouden. Uitzondering hierop is de herbestelling van een printplaat voor de scanner, dit was noodzakelijk doordat de ESP8266 enkel via complexe procedures HTTPS-verbindingen kon opzetten. Hierdoor was deze ESP dus ongeschikt voor ons project en is die vervangen door een ESP32. Hieronder een overzicht van alle gemaakte kosten (Tabel 1). De bestelde PCB's worden telkens per vijf geleverd.

Wanneer er gekeken wordt om EduBadge op grote schaal uit te rollen is dit voor een lage prijs mogelijk. Per scanner, dus per lokaal, is het 20,72 euro en op elke gang komt er een automaat. Deze kosten ongeveer 52,33 euro per stuk om te maken.

Tabel 1: Kostenberekening

Kostprijsberekening	
2 stuks ESP32-WROOM-32	12,98
Aramox Display Module, 2.8 Inch module 240 x 320 ILI9341 3,3 V SPI TFT	14,94
Drukknop OFF-(ON) Wit 1A – 125V	0,80
Diffused 5mm RGB LED CC	0,98
Step Motor 28BYJ-48 (5V) met ULN2003 driver	10,90
RFID Kit x2 stuks	10,44
PCB prototype pcb_automaat(ESP32), pcb_scanner(ESP8266)[2]	27,86
PCB prototype pcb_scanner(ESP32)	25,15
3D-prints	12,00
Binddraad 1,3 mm 40 m verzinkt	8,39
Piezo passieve buzzer	1,49
Totaal	**125,93**

[2] Bij het 1ste ontwerp van de scanner werd er een ESP8266 gebruikt. Echter kon deze zo goed als geen beveiligde verbinding aangaan waardoor er gekozen is om over te schakelen naar een ESP32 en er dus een nieuwe PCB moest besteld worden.

Conclusie

Het project EduBadge resulteert in de ontwikkeling en implementatie van een geautomatiseerd, RFID-gebaseerd aanwezigheidsregistratiesysteem. Dit systeem stelt studenten en docenten in staat hun aanwezigheden in specifieke lokalen efficiënt te beheren en te raadplegen, waarmee een significant verbeterd resultaat wordt behaald in het automatiseren van de aanwezigheidsprocessen binnen de opleiding Energietechnologie.

De realisatie van EduBadge omvat diverse cruciale deelaspecten. Op het gebied van hardware wordt een robuuste badge-automaat en scanner ontwikkeld. De automaat bestaat uit een RFID-lezer voor identificatie, een ESP32-microcontroller als centraal besturingssysteem en een touchscreen-display voor gebruikersinteractie. De scanner bestaat ook uit een RFID-lezer en ESP32 en beschikt daarnaast ook nog over een led en buzzer om een duidelijk visuele en auditieve feedback te geven. De software-architectuur omvat een gestructureerde database voor gebruikers-, lokaal- en aanwezigheidsgegevens, een backend die alle logica en communicatie tussen frontend en hardware afhandelt en een frontend voor gebruikersinteractie. De koppeling van studentenkaarten of tijdelijke badges aan gebruikersaccounts gebeurt via de automaat en via de website kunnen gebruikers zich registreren. De website biedt studenten de mogelijkheid om hun eigen aanwezigheden te raadplegen en hun badge te beheren (verwijderen), terwijl docenten een overzicht hebben van aanwezigheidsregistraties per lokaal of vak. Deze realisaties sluiten direct aan bij de eisen en beperkingen zoals geformuleerd in de inleiding. De eis van efficiënte en geautomatiseerde aanwezigheidsregistratie wordt volledig vervuld door de RFID-functionaliteit. De vraag naar een onderhoudsvrij en zo min mogelijk afhankelijk systeem wordt beantwoord door de keuze voor robuuste hardware en de interne logica van de applicatie. Hoewel geen vast budget is opgesteld, is de gekozen hardware en software gebaseerd op een kosteneffectieve aanpak. De beveiligingseisen zijn ingevuld door de implementatie van TOTP-authenticatie, e-mailverificatie, rolgebaseerde toegangscontrole en beveiligde communicatie met HMAC-tokens tussen de microcontrollers en de backend. Dit adresseert de oorspronkelijke zwakke punten van handmatige processen en onbeveiligde data-uitwisseling.

De praktische realisatie van EduBadge omvat de directe inzetbaarheid van het systeem in de Odisee Energietechnologie labo’s en de functionele beschikbaarheid van de bijbehorende webapplicatie. Dit resulteert in een onmiddellijke verlichting van de administratieve last voor docenten en een verbeterde gebruikservaring voor studenten. Voor mogelijke vervolgprojecten, aanpassingen en uitbreidingen zijn diverse opties denkbaar, waaronder integratie met bestaande onderwijssystemen, uitgebreide rapportagemogelijkheden, de ontwikkeling van een mobiele applicatie en de integratie van alternatieve authenticatiemethoden zoals QR-codes of biometrische gegevens mits de privacy en veiligheid gewaarborgd blijven. Deze projectresultaten en de potentiële uitbreidingen benadrukken het potentieel van EduBadge als een waardevolle oplossing voor efficiënte en veilige aanwezigheidsregistratie in onderwijsinstellingen.

Handleiding

Deze handleiding beschrijft de procedure om de EduBadge-aanwezigheidsregistratiedienst te configureren, te koppelen en te gebruiken.

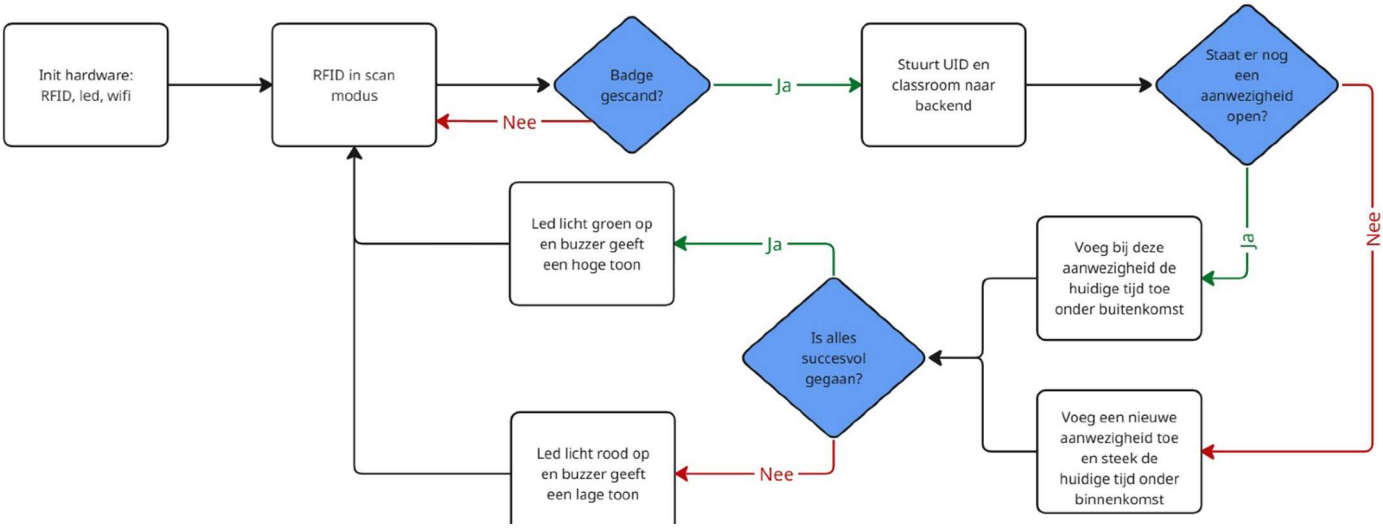
Navigeer naar de EduBadge-website. Klik door naar de registratiepagina en vul gebruikersnaam (voornaam.naam) en e-mailadres in. Maak een nieuwe gebruiker aan. Na succesvolle registratie wordt een QR-code of een TOTP-secretcode weergegeven. Scan de QR-code met een authenticatorapplicatie, zoals Google Authenticator, of voer het secret handmatig in de applicatie in. Activeer het account via de activatielink die wordt verstuurd naar het opgegeven e-mailadres.

Navigeer naar de inlogpagina van de website om in te loggen op het EduBadge-portaal. Voer het geregistreerde e-mailadres en de actuele 6-cijferige TOTP-code uit de authenticatorapplicatie in om toegang te verkrijgen tot het systeem.

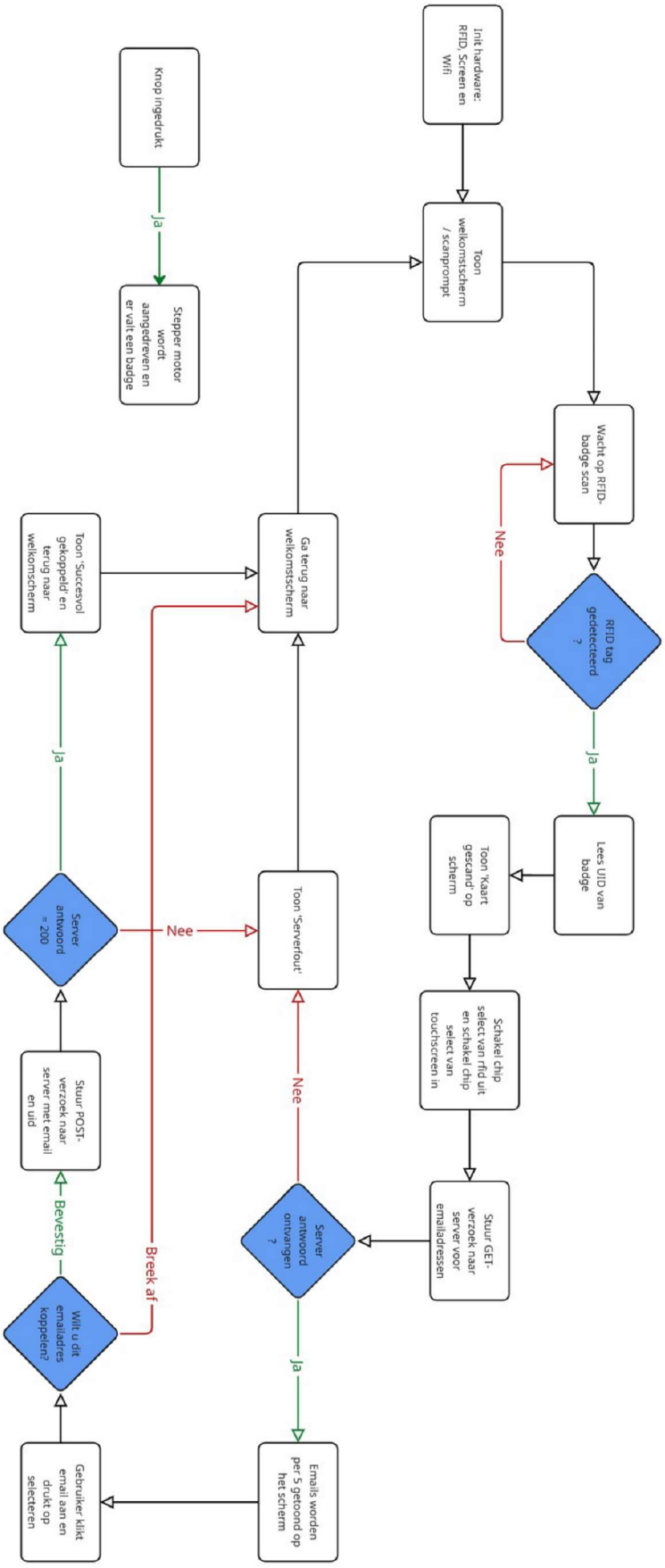
Scan de RFID-badge aan de daarvoor voorziene koppelautomaat om een badge te koppelen. Selecteer op het scherm van de automaat het bijbehorende e-mailadres en bevestig de keuze. Koppel op deze manier de badge aan het persoonlijke account. (**Fout! Verwijzingsbron niet gevonden.**)

Scan bij het betreden van een lokaal de badge aan de aanwezige scanner. Wanneer de buzzer klinkt en het indicatielampje groen oplicht, is de aanwezigheid correct geregistreerd. Brandt het lampje rood en klinkt er geen geluid, scan de badge dan opnieuw om een correcte registratie te verzekeren. Herhaal bij het verlaten van het lokaal hetzelfde proces door opnieuw de badge te scannen zodat het vertrek wordt geregistreerd. (Figuur 11)

Raadpleeg het persoonlijke aanwezigheidsoverzicht op de EduBadge-website door aan te melden. Ontkoppel hier een eventuele tijdelijke RFID-badge via het gebruikersdashboard van het account.



Figuur 11: Flowchart scanner



Figuur 12: Flowchart automaat

Literatuurlijst

[1] AZ-Delivery, *Handleiding voor de AZ-Delivery RFID RC522 Reader*. [Online]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/AZ216_A2-5_NL_B01M28JAAZ_ecaa464f-e1cf-48a9-88a0-7ddc4d767d84.pdf?v=1721036502

[2] AZ-Delivery, *Handleiding voor de AZ-Delivery ESP32 Dev kit*. [Online]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/AZ281_A_18-10_NL_B08BTWJGFX_f6c8a6af-23e5-4760-ae29-773d58fb01fd.pdf?v=1721128839

[3] "Amazon.com: ESP-WROOM-32 ESP32 ESP-32S Development Board 2.4GHz Dual-Mode WiFi + Bluetooth Dual Cores Microcontroller Processor Integrated with Antenna RF AMP Filter AP STA Compatible with Arduino IDE (1 PCS) : Electronics." https://www.amazon.com/ESP-WROOM-32-Development-Dual-Mode-Microcontroller-Integrated/dp/B07WCG1PLV?th=1

[4] Espressif Systems, "ESP32 datasheet," report, Oct. 2016. [Online]. Available: https://cdn.sparkfun.com/datasheets/IoT/esp32_datasheet_en.pdf

[5] AZ-Delivery, *2,8-inch TFT-scherm*. [Online]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/AZ406_D_21-01_NL_B0CDQ4LM39_9c3a8c50-891d-44ce-aa39-06979a2d5b84.pdf?v=1721132831

[6] "Wat is een Time-Based-One-Time-Password (TOTP) – Keeper," *Keeper® Wachtwoordbeheer & Digitale Kluis*. https://www.keepersecurity.com/nl_NL/resources/glossary/what-is-a-time-based-one-time-password/

[7] Formspree, "Easy HTML forms for static websites." [Online]. Beschikbaar: https://formspree.io/

[8] Geeksforgeeks, "Json web token."[Online].Beschikbaar: https://www.geeksforgeeks.org/json-web-token-jwt/

[9] Pencil & Paper, [Online]. https://www.pencilandpaper.io/articles/success-ux

Bijlagenoverzicht

Bijlage 1: Logboek rapporteren

Naam student	Paginanummers in rapport	Taak
Jethro	Hele document	Aanmaken
Mauro	Hele document	Aanmaken
Wout	Hele document	Aanmaken
Sander	Hele document	Aanmaken
Wout	6	Inleiding geschreven
Mauro, Jethro, Sander, Wout	6	Inleiding gecontroleerd
Mauro, Wout	Hele document	Aanpassen naar de gegeven feedback
Sander	10	Database geschreven
Mauro	10	Database nagekeken en verbeterd
Wout, Jethro	7-9	Hardwarecomponenten geschreven
Mauro	7-9	Hardwarecomponenten verbeterd
Mauro	Hele document	Figurenlijst en literatuurlijst gemaakt
Jethro,Sander, Wout	Hele document	Nagekeken en verbeterd
Sander	11-15	Frontend en backend geschreven
Mauro	Hele document	Aangegeven fouten verbeterd
Mauro	11	Database, backend inleiding en authenticatieproces geschreven
Mauro	17	ESP32 geschreven
Mauro	Hele document	Nagekeken en verbeterd
Wout	15, 16	ESP8266 geschreven
Wout, Jethro, Sander	Hele document	Nagelezen
Mauro	21	Handleiding geschreven
Jethro	15, 16	ESP8266 nagekeken en verbeterd
Jethro	17	ESP32 nagekeken en verbeterd
Mauro	15,16	ESP8266 herschreven naar scanner ESP32
Mauro	8	Wifirouter geschreven
Mauro	Begin, 24	Abstract en conclusie geschreven
Jethro	29-32	Hoofdstuk Ontwerpen geschreven
Jethro	3	Figurenlijst bijgewerkt
Wout	7	RFID-lezer tekst aangevuld

Wout	11, 12	Hoofdstuk Hardware schema's geschreven
Sander	17	Filteren en sorteren geschreven
Sander, Jethro, Wout	Hele document	Nagelezen
Mauro	Hele document	Opmaak goed gezet en nagelezen

Logboek_rapport.docx

Bijlage 2: Vergaderverslagen

- Verslag eerste vergadering project.docx
- Eerste evaluatiegesprek rapporteren.docx
- Verslag tweede vergadering project.docx
- Verslag derde vergadering project.docx
- Verslag vierde vergadering project.docx
- Tweede_evaluatiegesprek_rapporteren.docx
- Verslag vijfde vergadering project.docx
- Verslag zesde vergadering project.docx

Bijlage 3: Kopie van berekeningen, extra figuren

Datasheets

- RFID RC522:
  https://cdn.shopify.com/s/files/1/1509/1638/files/AZ216_A2-5_NL_B01M28JAAZ_ecaa464f-e1cf-48a9-88a0-7ddc4d767d84.pdf?v=1721036502

- ESP32:
  https://cdn.shopify.com/s/files/1/1509/1638/files/AZ281_A_18-10_NL_B08BTWJGFX_f6c8a6af-23e5-4760-ae29-773d58fb01fd.pdf?v=1721128839

- ESP32:
  https://cdn.sparkfun.com/datasheets/IoT/esp32_datasheet_en.pdf

- 2,4 inch TFT- kleurenschermmodule:
  https://cdn.shopify.com/s/files/1/1509/1638/files/AZ406_D_21-01_NL_B0CDQ4LM39_9c3a8c50-891d-44ce-aa39-06979a2d5b84.pdf?v=1721132831

- RGB led:
  https://www.gotron.be/media/files/Downloads/LED5RGBCC.pdf

- Stappenmotordriver:
  https://www.makerguides.com/wp-content/uploads/2019/04/ULN2003-Stepper-Motor-Driver-PCB.pdf

- Stappenmotor:
  https://www.makerguides.com/wp-content/uploads/2019/04/28byj48-Stepper-Motor-Datasheet.pdf

3D ontwerpen en printplaten

- Automaat-ESP32_pcb.brd
- Scanner-ESP32_pcb.brd
- Bottom_automaat.stl
- Top_automaat.stl
- Bottom_scanner.stl
- Top_scanner.stl
- Open-tool.stl