==

PROJECT PLAN DOCUMENT

Knowledge Kiosk Platform Upgrade

@Bosch Security Systems, Eindhoven, NL

==

TITLE	:	Knowledge Kiosk Platform Upgrade
DATE	:	February 14, 2025
VERSION	:	3.0
STATUS	:	Complete
AUTHOR	:	Sayem Ibne Taher

VERSION HISTORY

Version	Date	Author	Amendments	Status
1.0	14-02-2025	Sayem Ibne Taher	Initial draft of Project Plan	Draft
2.0	28-02-2025	Sayem Ibne Taher	Refined version	Complete

COMMUNICATION LOG

Version Sent	Date	Purpose	Stakeholders
1.0	14-02-2025	Shared for review	Frank Engelen
2.0	27-02-2025	Shared for feedback	Aljoscha Reuther

==

1. PROJECT ASSIGNMENT

==

1.1 CONTEXT & BACKGROUND

Organization Overview

Bosch Security and Safety Systems is a global leader in innovative solutions for safety and security, providing products and services for a wide range of industries. The ENG1 department within Bosch focuses specifically on the development of Public Address and Voice Evacuation products designed for environments with high safety requirements, such as airports, hospitals, cruise ships, shopping centers, and schools. These products are built to facilitate quick, reliable communication in emergencies, making safety accessible in high-traffic environments.

Relevant Developments

Within the engineering and R&D divisions, there has been a growing need for better internal communication and knowledge sharing to support innovation and ensure that staff have access to up-to-date information about new and ongoing developments. The recent creation of the Knowledge Kiosk platform is a direct response to this need, designed as an interactive, touch-based system located in common areas to facilitate on-demand information sharing among staff. However, with rapid growth in information and advancements in web and security technologies, the platform requires updates to improve user experience, accessibility, and security.

Current Situation

The Knowledge Kiosk Platform is an internal solution that provides employees and visitors access to announcements, training materials, and other internal resources via a touchscreen display. Currently, it is hosted on a local device running a basic Node.js/React.js application. The kiosk faces multiple challenges:

- Single-user login limitations (only one user can be logged in at a time)
- Inconsistent file upload process (frequent failures with PowerPoint/PDF files)
- Basic, and sometimes confusing, user interface
- Performance and security constraints due to local hosting and minimal Docker expertise

With the company aiming to improve employee engagement and streamline content management, **this project seeks to upgrade and optimise the kiosk** to handle multi-user access,better file uploads, **improved UI/UX**, and secure containerized deployment.

1.2 GOAL OF THE PROJECT

Address Current Pain Points: Solve single-user login limitation, improve the pdf or powerpoint slides uploading and displaying procedure and improve the UI for more intuitive navigation.
Enable Role-Based Access: Facilitate concurrent sessions (admin, user, viewer) for better flexibility.
Enhance Performance & reliability: Use Docker or other containerization methods to ensure consistent deployments and possible future expansion. Use version control tool and structures the documentation for better readability.

1.3 THE ASSIGNMENT

The assignment requires a new upgrade of the kiosk application by incorporating multi-user login with role-based access, refining the user interface and user experience (UI/UX), and implementing an better way of file uploading and displaying system (e.g: pptx/pdf). Additionally, the project must be containerized (using Docker or a suitable alternative) to simplify deployment and maintenance, and must include user/admin documentation, technical documentation, and test reports. If time and resources permit, explore and introduce other features that supports the aim of the knowledge kiosk.

1.4 SCOPE

Project Includes	Project Does not Include
Refactoring or building new authentication flows (multi-user, card-based login).	Full-scale hardware replacement (new touchscreen devices).
Upgrading file handling to accept PowerPoint/PDF with scheduling features.	Enterprise-wide hosting on multiple branches (initial focus on one kiosk location only).
Refactoring or building new authentication flows (multi-user, card-based login).	Paid third-party integrations or monthly subscriptions, unless approved.
Creating supporting documentation (admin/user guides, tech docs).	
Containerizing the kiosk app (Docker) for consistent deployments.	

1.5 CONDITIONS

Technology Set by the Company: Must respect security policies that limit hosting or third-party solutions.

Cost Constraints: The client strongly prefers free or one-time-cost services over recurring subscriptions.

Timeframe: 20 weeks

1.6 FINISHED PRODUCTS

Below is the Product Breakdown Structure (PBS) of deliverables:

Revised System Architecture Document
- Includes diagrams and explanations of authentication, file upload, Docker deployment.

Upgraded Kiosk Application
- Front end (React or Next.js with improved UI/UX).
- Back end (Node.js/TypeScript with multi-user authentication).
- Authentication (Token based JWT and mongodb database)
- Docker configuration and scripts (Dockerfiles, Docker Compose).
- An interactive design prototype (figma/sketch)

Research & Test Reports
- Summaries of user/stakeholder interviews, usability tests, performance tests.

Documentation
- Technical Documentation (architecture, APIs, container setups).
- Admin/User Guides (using the kiosk, uploading content, basic troubleshooting).

Final Presentation & Handover
- Presentation of outcomes, recommended next steps.

1.7 RESEARCH QUESTIONS

MAIN RESEARCH QUESTION

How can we upgrade the knowledge kiosk application for secured and efficient user authentication with reliable file uploading and management system?

SUB-QUESTIONS

How can we integrate multi-user login with role-based controls (viewer, uploader, admin)?

What methods can best handle PPT/PDF uploads for stable, on-time content display?

What design changes can reduce load times and improve the touchscreen user experience?

Which Docker practices can ensure consistent deployments and simplify maintenance?

==

2. APPROACH AND PLANNING

==

2.1 APPROACH

The project will follow an Agile/Scrum approach as discussed with the client, featuring:

Sprints: 2-week cycles for iterative development, with planning, reviews and retrospectives.

Stakeholder Reviews: Demos at the end of each sprint for feedback and priority alignment.

Weekly Stand-Ups: Short, frequent check-ins every week to address progress and challenges and coordinate tasks.

Completion Phase: Final testing, documentation, and presentation in the last sprint(s).

TEST APPROACH

Functional Testing: Predefined test cases for login, file upload, and UI flows.

Performance Testing: Load times, concurrency, and stress tests.

Usability Testing: With real or proxy kiosk users to validate UI/UX decisions.

Code Reviews: Ensuring quality and consistency in Node.js/React or Next.js code.

2.2 RESEARCH METHODS

DOT Framework with CMD Methods

This project follows the **Development Oriented Triangulation (DOT)** framework to structure research and communicate findings. It also incorporates **CMD methods** to support the project's scope and goals. At the same time, the project follows a **Human-Centred Design (HCD) approach**, moving through multiple phases and using different methodologies to achieve its objectives. The project consists of five phases: Empathise, Define, Ideate, Prototype & Test, and Develop. The choice of methodologies will vary based on the project's needs at each stage.

The HCD process is a user-focused approach to solving problems by deeply understanding user needs, brainstorming innovative solutions, and iterating based on feedback.

HCD Phase	CMD Methods	Complete	In-Progress	Incomplete
Empathise	Interviews, observation	Interviews, Observation	None	None
Define	User persona			Persona
Ideate	Proof of concept, Ideation			All
Protoype & Test	Sketch, A/B Testing, Heuristic Evaluation, Usability testing			All
Develop	Best, good and bad practices, Security Test, Quality review			All

2.3 LEARNING OUTCOMES

LEARNING OUTCOME 1: PROFESSIONAL DUTIES

By analyzing the current kiosk’s architecture, designing improved authentication and file management flows, and then realizing these changes in a Docker-based deployment, I will demonstrate professional-level IT tasks (analysis, design, realization, advising). The resulting deliverables such as revised application architecture documentation and a fully containerized application will represent industry-standard products characteristic of the kiosk’s IT domain.

LEARNING OUTCOME 2: SITUATION-ORIENTATION

Applying previously gained React/Node/Database management and UI/UX knowledge within the company's real operational context, I will deliver relevant outcomes for stakeholders. By adapting my method of work (Agile sprints) to the company's processes, I will ensure the final solution directly addresses and benefits the client's actual needs.

LEARNING OUTCOME 3: PROFESSIONAL STANDARD

I will take ownership of resolving the kiosk's issues, using research to inform decisions regarding multi-user authentication, secure file handling, docker deployment, and potential expansion. In a setting with uncertainties (infrastructure limits, security policies), I will justify my chosen approaches with findings from technical studies, always bearing in mind ethical and sustainable considerations for the long-term usability of the kiosk solution.

LEARNING OUTCOME 4: PERSONAL LEADERSHIP

By setting my own goals in learning and applying docker containerization, optimizing concurrency, efficient file upload and management with proper application error handling, and refining user interfaces will reflect my initiative and adaptability. Reflecting regularly on my development and seeking feedback from internship coach and company mentor showcases my capacity to lead myself, align with my learning objectives with project needs, and evolve as an IT professional.

2.4 BREAKDOWN OF THE PROJECT (Estimated)

Below is a high-level breakdown by sprints ***(approx. 2 weeks each as recommended by the client considering a timeline of 20 weeks. If one task is completed before the sprint deadline a new task will take place and if one task is delayed it will automatically move to the next sprint.)***

Sprint 0 (Preparation)

I set up the development environment, conduct stakeholder interviews, and gather detailed requirements to ensure clarity on the project objectives and constraints.

Sprint 1

I focus on the initial authentication setup, investigate the card reader integration, begin learning Docker fundamentals, and update relevant documentation and my portfolio.

Sprint 2

I refine the file upload feature for PPT/PDF, finalize the core requirements, and establish an initial Docker configuration for testing.

Sprint 3
Create wireframes and validate them with the client, research memory management strategies, and continue to update documentation and my portfolio.

Sprint 4
Conduct usability testing, fix newly discovered bugs, develop a high-fidelity prototype, and enhance documentation to reflect the latest changes.

Sprint 5
Perform A/B testing to finalize the design and initiate core development tasks, ensuring that documentation remains current.

Sprint 6
Continue development , integrating and refining features while updating documentation to capture the project’s progress..

Sprint 7
Focus on additional development, bug fixing, application-wide testing, and design consistency checks, all while keeping documentation up to date.

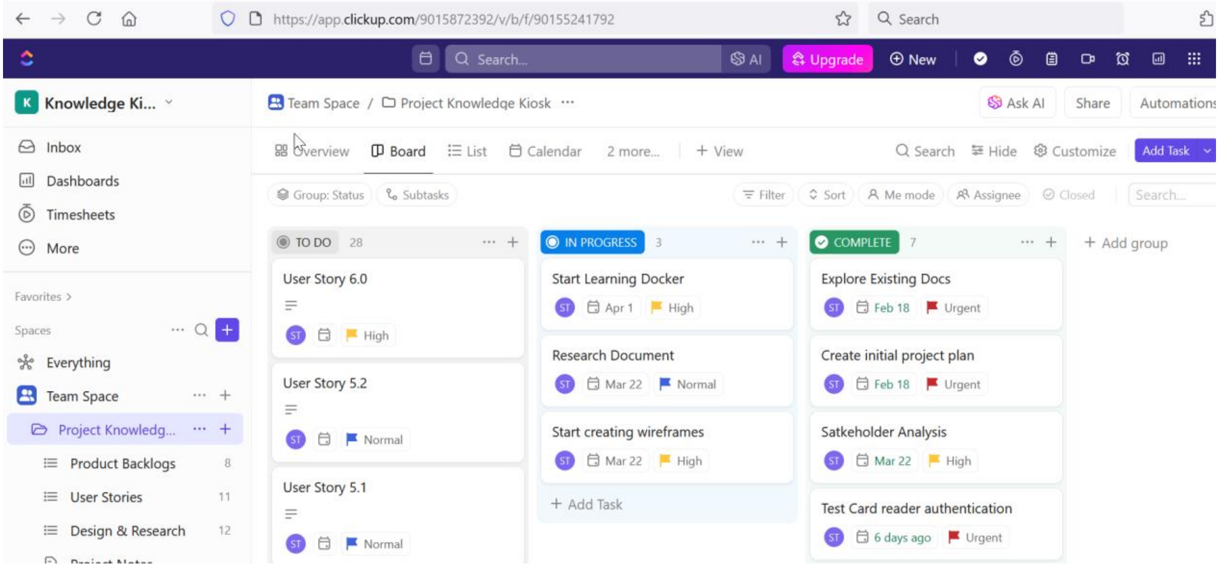
Sprint 8
Address hosting and deployment requirements, apply any final bug fixes, and ensure the deployment plan aligns with the project’s constraints.

Sprint 9 (Closure)
Develop and finalize the user manual, complete all documentation, hand over the final deliverables, and prepare for the final presentation.

2.5 TIME PLAN (Estimated)

Sprints	Number of Weeks	Start Date	Completion
Sprint 0 (preparation)	2	February 3rd, 2025	February 16th, 2025
Sprint 1	2	February 17th, 2025	March 2nd, 2025
Sprint 2	2	March 3rd, 2025	March 16th, 2025
Sprint 3	2	March 17th, 2025	March 30, 2025
Sprint 4	2	April 1st, 2025	April 14th, 2025
Sprint 5	2	April 15th, 2025	April 28th, 2025
Sprint 6	2	April 29th, 2025	May 12th, 2025
Sprint 7	2	May 13th, 2025	May 26th, 2025
Sprint 8	2	May 27th, 2025	June 9th, 2025
Sprint 9 (closure)	2	June 10th, 2025	June 24th, 2025

A comprehensive overview of planning and approaches can be found here



TO DO 28

- User Story 6.0 — ST — High
- User Story 5.2 — ST — Normal
- User Story 5.1 — ST — Normal

IN PROGRESS 3

- Start Learning Docker — ST — Apr 1 — High
- Research Document — ST — Mar 22 — Normal
- Start creating wireframes — ST — Mar 22 — High
- + Add Task

COMPLETE 7

- Explore Existing Docs — ST — Feb 18 — Urgent
- Create initial project plan — ST — Feb 18 — Urgent
- Satkeholder Analysis — ST — Mar 22 — High
- Test Card reader authentication — ST — 6 days ago — Urgent

==

3. PROJECT ORGANISATION

==

3.1 TEAM MEMBERS

Name	Role	Availability	Contact
Sayem Ibne Taher	Designer & Developer (The Intern)	Mon - Fri (9:00 - 17: 00)	+31687001745 fixed-term.sayem.taher@nl.bosch.com
Frank Engelen	Company mentor and Product owner	Mon - Fri (9:00 - 17: 00)	frank.eneglen2@nl.bosch.com
Nick Hoogland	Additional company mentor	Mon - Fri (9:00 - 17: 00)	nick.hoogland@nl.bosch.com

3.2 COMMUNICATION

- Weekly Stand-Up: Quick updates with supervisor.
- Bi-Weekly Sprint Review & Retro: Show demos to stakeholders, gather feedback.
- Bi-Weekly Internship Coach Check-In: Align with Fontys teacher on progress and learning outcomes.
- Ad Hoc: Emails/MS Teams messages for urgent items.

3.3 TEST ENVIRONMENT

- Local Dev Environment: Node.js, React/Next.js with Vite, Docker on local machine, MongoDB Database on local machine.
- CI/CD Pipeline: If possible, Git-based automated builds and basic tests.

3.4 CONFIGURATION MANAGEMENT

- Git Repository with Branching Strategy (main, develop, feature branches).
- Baselines & Releases: Each sprint produces a "release candidate."

3.5 MAINTENANCE

- Handover the application as a ZIP File so that the main code files remains intact in case the git repository is not accessible
- The Figma prototype link will be shared so that the interactive prototype can be checked.
- Documentation Storage: Microsoft Teams

==

4. FINANCE AND RISKS

==

4.1 RISKS AND FALL- BACK ACTIVITIES

Risk	Description	Probability	Impact	Mitigation Strategy
Limited Hosting Options	Company server restrictions may prevent certain deployment methods or expansions.	High	High	Propose on-premises Docker deployment; investigate minimal-cost solutions that comply with security rules. If not possible for third party integration the application server will run locally.
Concurrent Login Complexity	Implementing multi-user authentication with card-based login may introduce new bugs.	Medium	Medium	Thoroughly plan data flow and session management; perform iterative testing and code reviews.
Lack of Docker Expertise	Inexperience with Docker may slow development and cause deployment issues.	Medium	Medium	Get training resources, adopt well-documented setups, allocate time for trial runs.
Hardware Limitations	Touchscreen device may not support multi-touch or have hardware constraints.	Low	Medium	Test early to confirm device limitations; design UI for single-touch fallback if needed.
Scope Creep	Changes to requirements or new requests mid-project could derail timeline.	Medium	High	Maintain a clear change-control process; get stakeholder sign-off before accepting new features.

==

5. OTHER

==

- Ethical & Sustainable Choices: Reuse existing kiosk hardware to minimise waste, ensure forward-looking design for possible multi-branch support.
- Personal Development: Track reflections on Docker, concurrency, multi-user security; align with career goals in IT.

==

END OF PROJECT PLAN DOCUMENT

==