TIE 204: MVP Studio

Course Information

Offering Department: Engineering

Academic Semester: Spring 2025

Instructor: Prof. Dr. Amr T. Abdel-Hamid

Email: amr.abdelhamid@kaust.edu.eg

Class Schedule: Weekly Sessions (1-hour lecture + 2-hour lab)

Course Description

TIE 204 builds upon the foundation laid in TIE 203 by guiding students through the transformation of working prototypes into secure, optimized, and scalable MVPs (Minimum Viable Products). The course emphasizes advanced system integration, performance optimization, security, edge AI deployment, and real-world validation. Students will engage in rigorous testing, stakeholder engagement, and cross-disciplinary collaboration to ensure their designs are viable, sustainable, and ready for real-world impact.

Learning Outcomes

Upon successful completion of this course, students will be able to:

- Optimize and refine prototypes to meet real-world constraints and performance goals.
- Apply secure design principles and implement robust communication protocols.
- Design and integrate AI capabilities at the edge and within backend systems.
- Conduct stakeholder engagement, validation testing, and iterate based on feedback.
- Communicate technical solutions effectively to technical and non-technical audiences.

Course Structure

TIE 204 is organized into four sprints that guide students from problem understanding to a deployable MVP, following real-world product development practices.

- Sprint 1 – Problem & System Definition (Weeks 1–3)
    - Focus on validating the problem and designing the system before any implementation.
    - Problem validation and user context
    - Solution exploration and algorithm intent
    - System architecture (HW + SW boundaries)

    - Design rules, interfaces, and test thinking
      Outcome: Design-ready system concept and architecture freeze (v1)

- Sprint 2 – Architecture & Prototyping (Weeks 4–6)
    - Focus on turning architecture into a working prototype.
    - Hardware and software architectural realization
    - Early schematic and system layout
    - Integration planning and MVP alpha build
      Outcome: MVP Alpha with validated architecture

- Sprint 3 – Implementation & Optimization (Weeks 7–9)
    - Focus on making the system robust and efficient.
    - Embedded and system security concepts
    - Unit testing and system testing strategies
    - Local optimization and performance tuning
      Outcome: Functional MVP with tested core features

- Sprint 4 – Validation, Scaling & Readiness (Weeks 10–14)
    - Focus on preparing the MVP for real-world use and demonstration.
    - CI/CD and iterative deployment practices
    - System validation and benchmarking
    - UX refinement, documentation, and public demo
      Outcome: Final MVP, technical documentation, and demo-ready system

Weekly Schedule

Phase 0 – Shared Foundations (Weeks 1–3)

In this phase, all teams follow the same activities. No implementation is allowed. The focus is on problem understanding, algorithms, and system architecture.

Week	Focus	Lecture Theme	Studio Focus (Everyone)	Deliverables
1	Foundation	Course intro, FIRE, MVP mindset	Problem framing, system sketch	
2	System Thinking	System design, algorithms, architecture	System boundaries, Algorithm Intent Tables	Architecture Sketch, Algorithm Table
3	Design Readiness	Design rules, interfaces, **testing mindset**	Architecture refinement, design review	Design Freeze v1, Spec Draft

Phase 1 – Architecture & Prototyping (Weeks 4–6)

From this phase onward, projects diverge in implementation, but all teams must satisfy the same engineering questions and milestones.

Week	Sprint	Lecture Theme	Project Work (Team-Specific)	Deliverables
4	Sprint 2	System architecture → circuits & SW structure	HW: schematic concepts | SW: system layout	Schematic v1, System Blueprint
5		Constraints & trade-offs	HW: layout constraints | SW: UI / API logic	PCB Draft, UI Wireframes
6		Integration mindset	HW/SW integration planning	**MVP Alpha Snapshot**

Phase 2 – Implementation & Optimization (Weeks 7–9)

In this phase, depth depends on the nature of the project. Not all projects perform identical tasks, but all must justify design decisions.

Week	Sprint	Lecture Theme	Project Work	Deliverables
7	Sprint 3	Security by design	HW: Physical security | SW: authentication	Security Plan
8		Testing in systems	HW: measurements | SW: unit tests	Test Checklist
9		Local optimization	Calibration / performance tuning	**Functional MVP**

Phase 3 – Validation, Scaling & Presentation (Weeks 10–14)

Week	Sprint	Lecture Theme	Project Work	Deliverables
10	Sprint 4	CI/CD & iteration	Firmware / service pipelines	CI/CD Evidence
11		System validation	Rig tests / integration tests	Validation Plan
12		Benchmarking	Power, latency, load (as applicable)	Performance Summary
13		Product polish	UX, enclosure, documentation	Final Build Package
14		Reflection & demo	Public demo & feedback	**Final MVP, Poster, Tech Docs**

Key Message to Students

Weeks 1–3 are fully synchronized across all teams.
From Week 4 onward, projects diverge in implementation, but engineering standards remain fixed.

You are evaluated on reasoning, architecture quality, and engineering decisions — not on identical outputs.

Assessment Strategy

Assessment Component	**Weight**
Sprint 1 – Foundation & Specification	10%
Sprint 2 – Architecture & Prototyping	10%
Sprint 3 – Implementation & Optimization	10%

Final MVP Demo & Documentation	40%
Quizzes (3 total – Best 2 counted)	15%
Final Exam (Technical Concepts & System Design)	10%
Team Reflection & Peer Assessment	5%
Total	**100%**

Required Materials

- ESP32/XIAO microcontrollers, edge AI-capable devices
- Cloud platforms: ThingsBoard, MQTT, OTA update tools
- Tools: TensorFlow Lite, GitHub, CI/CD pipelines

Sustainability and SDG Alignment

This course supports **Goal 4**: Quality Education by immersing students in an advanced, practice-driven studio where they transform functional prototypes into secure, scalable MVPs. Through optimization, edge AI, and deployment readiness, it builds deep technical and collaborative skills. It addresses **Goal 11**: Sustainable Cities and Communities **and Goal 13**: Climate Action through energy-efficient designs, predictive maintenance, and smart infrastructure applications. It also supports Goal 17: Partnerships for the Goals by encouraging industry collaboration, stakeholder validation, and scaling of real-world solutions.

Course Policies

- Attendance and full participation in at least 80% of labs and presentations are mandatory.
- Projects are team-based, with shared ownership and peer accountability.
- All final projects must demonstrate real-world readiness and validation.