CAMPUS BOOKING SYSTEM

Inception Document

Phase 1 · v1.0 · CBS-INC-001

Purpose of This Document

This Inception Document establishes the foundational understanding of the Campus Booking System (CBS) project. It captures the business context, problem statement, project scope, key stakeholders, high-level risks, and the rationale for the chosen prototype approach. It is the authoritative starting point from which all subsequent requirements, design, and development work is derived.

Section 1 — Business Context & Background

The university identified the following systemic gaps:

No unified digital platform for booking any campus resource

Tutor scheduling dependent entirely on manual coordination

Room availability communicated via paper notices or phone calls

Equipment lending tracked through physical sign-out sheets

Zero notification infrastructure for reminders, cancellations, or waitlists

Administrators lack data-driven visibility into resource utilisation

The CBS project was commissioned to resolve these gaps through a mobile-first SwiftUI iOS prototype that validates core booking flows before full production investment.

Section 2 — Project Objectives

The CBS project has three primary objectives for Phase 1:

Section 3 — Scope Definition

In Scope — Phase 1

SwiftUI iOS prototype for all 7 epics

Mock data simulating authentication, bookings, notifications, admin, and ratings

Full requirements documentation with traceability to ISO/IEC 25010

Document control infrastructure (policy, procedure, logs)

Vision, mission, and feature set documentation

Landing page linking all Phase 1 artifacts

Out of Scope — Phase 1

Live backend API or database integration

Production App Store deployment

Real-time calendar or university system integrations

Payment processing or fee management

Multi-institution or multi-campus support

Section 4 — Stakeholder Register

Section 5 — Assumptions & Constraints

Assumptions

University students have access to iOS 17+ devices

Mock data is sufficient to validate prototype flows without a live backend

Team members are available for the full Phase 1 sprint cadence

All booking policies (duration limits, cancellation windows) will be defined by the Product Owner

Accessibility standards (WCAG 2.1 AA) will be validated during design review, not automated testing

Constraints

Prototype must be buildable within the Phase 1 project timeline

Technology stack is fixed: Swift 5.9+, SwiftUI, iOS 17+, Xcode 15+

No external APIs or paid services may be integrated in Phase 1

All documentation must follow the CBS document control policy (CBS-POL-DC-001)

Team size is fixed at 5 members with defined roles

Section 6 — Risk Register

Section 7 — Prototype Approach & Rationale

The decision to deliver a SwiftUI iOS prototype rather than a full production application is based on the following strategic rationale:

Validated learning over premature investment — A prototype allows the team to test whether the core booking flows meet real user needs before committing to a full backend build.

Mobile-first appropriateness — University students are predominantly mobile users. An iOS-native prototype delivers the most realistic representation of the target experience.

MVVM architecture alignment — The prototype's MVVM structure is designed to translate directly to a production codebase, reducing rework in Phase 2.

Risk reduction — A scoped prototype limits technical and financial risk while generating high-value documentation and design artefacts.

Stakeholder communication — A navigable iOS prototype is a more effective stakeholder communication tool than static wireframes or written specifications alone.

Section 8 — Phase 1 Deliverables Summary

Document ID: CBS-INC-001

Version: 1.0

Product Owner: Jason Myers

Phase: Phase 1

The Campus Booking System addresses a fragmented and inefficient resource access landscape at the university. Students, tutors, and administrative staff currently rely on disparate manual systems, email chains, and departmental front desks to access study rooms, equipment, and tutoring support.

Objective	Description	Success Measure
OBJ-1	Deliver a working SwiftUI iOS prototype covering all 7 epics with mock data	All 7 epics have at least one navigable screen
OBJ-2	Produce controlled, versioned documentation aligned to ISO/IEC 25010	All docs carry IDs, versions, and approvals
OBJ-3	Validate core booking user flows for Student, Tutor, and Admin personas	End-to-end flows are demonstrable in the prototype

Stakeholder	Type	Interest	Influence
Student Users	Internal	Book resources easily on mobile	High
Tutor Users	Internal	Manage availability and session bookings	Medium
Admin Users	Internal	Oversee resources and approvals	High
University IT Department	Internal	Ensure technical standards compliance	High
Facilities Management	Internal	Optimise room and equipment utilisation	Medium
Academic Services	External	Ensure equitable and accessible service delivery	Medium
Compliance Team	Internal	Verify audit trails and data protection compliance	Medium

Risk	Description	Likelihood	Impact	Mitigation
Scope Creep	Team adds features beyond Phase 1 scope	Medium	High	Strict MoSCoW gating by Product Owner
SwiftUI Complexity	Novel SwiftUI patterns delay delivery	Medium	Medium	Tech Lead provides architecture guidance upfront
Stakeholder Misalignment	Requirements change after baseline	Low	High	Requirements baseline signed off before development
Team Availability	Member unavailability affects sprint delivery	Low	Medium	Scrum Master tracks velocity and redistributes tasks
Documentation Gaps	Missing or uncontrolled documents fail audit	Low	High	Document control policy enforced from Sprint 0

Doc ID	Deliverable	Owner	Status
CBS-INC-001	Inception Document	Jason Myers	Complete
CBS-VIS-001	Vision Document	Jason Myers	Complete
CBS-MIS-001	Mission Document	Jason Myers	Complete
CBS-FEAT-001	Feature Set Document	Umaya Hassan	Complete
CBS-IDEA-001	Idea Generation Declaration	Bassel Taleb	Complete
CBS-ELIC-001	Elicitation Requirements	Umaya Hassan	Complete
CBS-ELAB-001	Elaboration / NFR Document	Saad Alzamzami	Complete
CBS-POL-DC-001	Document Control Policy	Jason Myers	Complete
CBS-PROC-DC-001	Document Control Procedure	Saad Alzamzami	Complete
CBS-LOG-DC-001	Document Control Logs	Bassel Taleb	Complete
CBS-REQDB-001	Requirements Database	Umaya Hassan	Complete
CBS-PROTO-001	SwiftUI iOS Prototype	Saad Alzamzami / Bassel Taleb	In Progress