Product Requirements Document (PRD)

Project Name: University Connect

Date: June 23, 2025

1. Executive Summary

Objective: Build a cross-platform mobile application using Flutter (frontend/UI) and Firebase (backend) to enhance university experience through secure authentication, calendar sync, personalized notifications, complaint management, QR-based access & attendance, campus navigation, and exam/grade alerts.

Key Features:

1. Microsoft OAuth Login
2. Academic Calendar Integration
3. Personalized Notifications
4. Complaint Management
5. QR Code Campus Access
6. QR Attendance System
7. Campus Navigation
8. Exam & Grade Notifications

2. Goals & Success Metrics

- **Adoption Rate:** 60% of students & staff active within first semester
- **Authentication Uptime:** ≥99.9%
- **Notification Engagement:** 70% of push notifications acknowledged
- **Complaint Resolution Time:** Average ≤ 48 hours
- **Attendance Accuracy:** ≥98% QR scans correctly recorded

3. Stakeholders

- **Product Owner:** University IT Director
- **Project Manager:** Assigned by dev team
- **Development Team:** Flutter developers, Firebase engineers
- **QA & Security:** QA lead, InfoSec officer
- **End Users:** Students, Faculty, Administrative Staff

4. User Personas

Persona	Role	Goals	Pain Points
Undergraduate	Student	View grades, track schedule, navigate campus	Missed campus events, late alerts, complex maps
Professor	Academic	Post announcements, track attendance, manage issues	Manual attendance logs, complaint tracking overhead
Admin Staff	Administrator	Resolve complaints, publish events	Dispersed feedback channels, calendar updates delay

5. User Stories & Functional Requirements

5.1 Microsoft OAuth Login

- **User Stories:** As a user, I want to log in with my Microsoft university account so that I don’t need a separate password.
- **Requirements:**
- Integrate Microsoft Azure AD OAuth2.0
- Single Sign-On (SSO)
- Role extraction (student vs. staff)
- Secure token storage (Firebase Auth)

5.2 Academic Calendar Integration

- **User Stories:** As a user, I want the app to sync my university calendar so I can see upcoming classes and events.
- **Requirements:**
- Consume university’s calendar API (iCal/REST)
- Two-way sync: read events; optional personal-event injection
- Offline caching (Firestore)

5.3 Personalized Notifications

- **User Stories:** As a user, I want to receive alerts relevant to my role and preferences.
- **Requirements:**
- Firebase Cloud Messaging (FCM) for push
- Topic-based subscriptions (e.g. "CS101 Alerts", faculty announcements)
- In-app preferences panel for notification types and quiet hours

5.4 Complaint Management

- **User Stories:** As a student, I want to submit and track a complaint regarding facilities or courses.
- **Requirements:**
- Complaint submission form (text, file upload)

- Status lifecycle (Submitted → In Review → Resolved)
- Admin dashboard for review & resolution
- Email notifications on status change

5.5 QR Code Campus Access

- **User Stories:** As a student, I want to scan a QR code to enter buildings securely.
- **Requirements:**
- Unique time-bound QR per user (PDF417/QR code)
- Campus scanners validate via Firebase Functions
- Access logs stored in Firestore

5.6 QR Attendance System

- **User Stories:** As a professor, I want students to scan a QR at class start to mark attendance.
- **Requirements:**
- Session-specific QR code generation
- Real-time attendance logging (Firestore)
- Attendance summary reports (CSV export)

5.7 Campus Navigation

- **User Stories:** As a newcomer, I want an interactive map showing buildings, routes, and live shuttle locations.
- **Requirements:**
- Map widget (Google Maps SDK / OpenStreetMap)
- GeoJSON layers for building polygons
- Real-time shuttle GPS via REST sockets
- Route planner (walking, shuttle)

5.8 Exam & Grade Notifications

- **User Stories:** As a student, I want instant alerts for exam schedules and grade releases.
- **Requirements:**
- Consume exam/event API
- FCM for exam reminders and grade push
- Secure grade retrieval via Firebase Cloud Functions

---

6. Non-Functional Requirements

- **Performance:** <2s screen load, <1s QR validation
- **Scalability:** Support 20k concurrent users
- **Security:** OWASP mobile best practices, data encryption at rest & in transit
- **Availability:** 99.9% SLA, automated backups
- **Localization:** English + Turkish support
- **Accessibility:** WCAG AA compliance

---

7. Technical Architecture

1. **Frontend:** Flutter (Dart) – Android & iOS
2. **Backend:** Firebase Authentication, Firestore, Cloud Functions, Cloud Messaging, Storage
3. **Third-Party APIs:** Microsoft Azure AD, University Calendar & Exam APIs, Map SDK
4. **CI/CD:** GitHub Actions → Firebase App Distribution → Play/App Store

Architecture Diagram:

```
[Client Apps] ↔ [Firebase Auth] ↔ [Microsoft OAuth]
              ↕
           [Firestore] ← [Cloud Functions] → [Uni APIs]
              ↕
         [Cloud Messaging]
```

---

8. Data Model (High-Level)

- **Users:** uid, name, email, role, photoURL, notificationPrefs
- **Events:** eventId, title, description, start, end, type
- **Complaints:** complaintId, userId, category, description, attachments, status, timestamps
- **AccessLogs:** logId, uid, locationId, timestamp
- **Attendance:** recordId, sessionId, uid, timestamp
- **MapLocations:** locationId, name, coords, type

---

9. API & Integration Details

Feature	API Owner	Protocol	Auth
Microsoft OAuth	Azure AD	OAuth2.0	JWT (Bearer)
Academic Calendar	University IT	REST/iCal	API Key + OAuth
Exam & Grade Info	University IT	REST	OAuth2.0
Shuttle Tracking	Transport Team	WebSocket	API Key

---

10. UX/UI & Wireframes

- **Login Screen:** Microsoft SSO button, role icon
- **Home Dashboard:** Upcoming events, unread notifications
- **Map View:** Search bar, layer toggles, real-time shuttle pins
- **Complaint Form:** Category dropdown, attachments gallery

(Detailed wireframes to be designed in Figma)

11. Roadmap & Milestones

Phase	Duration	Deliverables
Discovery	2 weeks	Detailed specs, final wireframes
Alpha	4 weeks	Auth, Calendar, Notifications
Beta	6 weeks	Complaints, QR Access & Attendance, Map
Release Candidate	3 weeks	Exam/Grade, Localization, QA & Security audit
Launch	2 weeks	Store submissions, Marketing assets

12. Risks & Mitigations

- **API Reliability:** Schedule regular stubs/fallbacks
- **SSO Failures:** Implement retry & offline Mode
- **Data Privacy:** GDPR/TKVK compliance review
- **User Adoption:** Partner with student ambassadors

13. Appendix

- Glossary of Terms
- Contact List
- Compliance Checklist

End of PRD

Features Document: University Connect

Below is a granular breakdown of every feature, describing purpose, user flows, acceptance criteria, and dependencies.

1. Microsoft OAuth Login

Description: Enable users to authenticate with their official Microsoft university accounts for secure SSO. **User Flow:**

1. User taps “Login with Microsoft.”
2. Redirect to Azure AD consent screen.
3. User consents & grants permissions (basic profile, email).
4. App receives OAuth token & exchanges for Firebase Auth credentials.
5. User lands on dashboard with role-based UI.

Acceptance Criteria:

- Microsoft button present on login screen.
- Successful token exchange 100% of test scenarios.
- Role claim (student/staff) correctly parsed and UI elements toggle accordingly.
- Error messages displayed for network failures or invalid credentials.

Dependencies: Microsoft Azure AD tenant setup, Firebase Auth configuration.

2. Academic Calendar Integration

Description: Sync university’s master calendar, letting users view and optionally subscribe to events. **User Flow:**

1. On first login, app prompts to sync calendar.
2. User consents; app fetches events via REST/iCal.
3. Events stored in Firestore & displayed in calendar view.
4. User can add custom reminders or personal entries.

Acceptance Criteria:

- Automated daily sync job updates events within 5 minutes of changes.
- Calendar view displays events with proper color-coding by type.
- Custom event creation persists locally and in user’s personal collection.

Dependencies: University calendar API endpoint.

3. Personalized Notifications

Description: Tailored alerts pushed to users based on role, course enrollment, and preferences. **User Flow:**

1. Onboarding directs user to notification settings.
2. User selects topics (e.g., Grades, Events, Complaints).
3. FCM topics subscribed/unsubscribed accordingly.
4. Notifications appear in-app and as push.

Acceptance Criteria:

- Topic toggles update FCM subscriptions in real time.
- Notifications correctly routed to intended user roles.
- Quiet hours honored; no alerts sent during configured hours.

Dependencies: Firebase Cloud Messaging, Firestore prefs document.

4. Complaint Management

Description: Comprehensive system for lodging, tracking, and resolving complaints. **User Flow:**

1. User opens “Complaints” tab and fills form (category, description, attachments).
2. Submission triggers Firestore document creation.
3. Admin sees new case in dashboard, assigns status.
4. Status updates (In Review, Resolved) notify user via email & in-app.
5. Closed complaints archived.

Acceptance Criteria:

- File uploads (images, PDFs) successfully stored in Firebase Storage.
- Status transitions enforce correct sequence (Submitted → In Review → Resolved).
- Timestamp logs visible in complaint history for audit.

Dependencies: Firebase Storage, Cloud Functions for email.

5. QR Code Campus Access

Description: Generate dynamic QR codes for secure door access. **User Flow:**

1. User taps “Access Pass”; app calls Cloud Function to generate time-bound QR.
2. QR displayed with countdown timer valid for 5 minutes.
3. Scanner device reads QR, validates token via Firebase Functions, and grants/denies entry.
4. Access event logged.

Acceptance Criteria:

- QR refreshes automatically every 5 minutes.
- Invalid/expired QR returns appropriate error.
- Access logs include uid, timestamp, locationId.

Dependencies: Campus door scanner integration, Cloud Functions.

6. QR Attendance System

Description: Mark attendance by scanning per-session QR codes. **User Flow:**

1. Professor generates session QR in “Attendance” screen.
2. Students scan QR; each scan logs attendance with Firestore timestamp.
3. Professor reviews attendance roster in-app or exports CSV.

Acceptance Criteria:

- Only one scan per user per session accepted.
- Late scans beyond session window (e.g., 10 minutes) flagged.
- CSV export contains uid, name, timestamp, sessionId.

Dependencies: Firestore, Cloud Functions for export.

7. Campus Navigation

Description: Interactive map guiding users across campus, including live shuttle tracking. **User Flow:**

1. User opens “Map” tab showing campus overlay.
2. User searches or taps a building to get info and directions.
3. Toggle shuttle layer to see live bus icons.
4. Route planner offers walking/shuttle options.

Acceptance Criteria:

- Map loads within 3 seconds, with building polygons from GeoJSON.
- Shuttle positions update every 30 seconds.
- Directions accurately plot on map with turn-by-turn text.

Dependencies: Google Maps/OpenStreetMap SDK, shuttle GPS feed.

8. Exam & Grade Notifications

Description: Push alerts for upcoming exams and grade postings. **User Flow:**

1. Exams imported alongside calendar events with reminder flags.
2. Grade releases trigger Cloud Function that sends FCM to enrolled students.
3. Students tap notification to view results in secure grade view.

Acceptance Criteria:

- Exam reminders sent 24h and 1h before event start.
- Grade notifications link to grades page; unauthorized access locked.
- Audit logs for notification delivery status.

Dependencies: Exam & grade REST API, Cloud Functions.

End of Features Document