**2.1 Context Objects & Requirement Sources**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Context Object** | **Why Relevant** | **Primary Requirement Sources** |
| C1 | **Student / Staff Rider** | End users who request or accept rides | Pre project survey and semi structured interviews |
| C2 | **Driver** | Provides vehicle seats; needs parking privileges | Focus group workshop notes |
| C3 | **Campus Parking Space** | Real time slot reservation and fee data | Smart parking IOT DB & facilities manager interview ([Animo Repository](https://animorepository.dlsu.edu.ph/cgi/viewcontent.cgi?article=2419&context=conf_shsrescon&utm_source=chatgpt.com)) |
| C4 | **Vehicle Registry** | Plate to user mapping for campus security | Campus security policy documents |
| C5 | **Digital Campus SSO** | Trusted login, delivers user roles | LDAP‑over‑TLS best‑practice doc ([calnet.berkeley.edu](https://calnet.berkeley.edu/calnet-technologists/ldap-directory-service/ldap-best-practices?utm_source=chatgpt.com)) |
| C6 | **In‑App Wallet** | Splits ride cost & parking fees | Finance Dept SLA |
| C7 | **Campus Map & Sensors** | Provides traffic and ETA data | GIS layer specification |
| C8 | **University Safety Office** | Receives SOS alerts, incident logs | Safety SOP |
| C9 | **Road Transport Regulations** | Legal boundary for non commercial carpooling | Malaysia e‑hailing regulation summary ([Asia Law Portal](https://asialawportal.com/regulating-e-hailing-in-malaysia-is-there-over-regulation/?utm_source=chatgpt.com)) |
| C10 | **Benchmark Apps** (Grab, Kumpool) | UX baseline & delighters | Competitive heuristic teardown |

**3.1 Functional Requirements(FR)**

|  |  |  |
| --- | --- | --- |
|  | Requirement | Rationale |
| FR‑1 | The system shall authenticate every user through the university Single Sign‑On (SSO) service before any other function is available. | Secure access is a baseline expectation ([NIST Computer Security Resource Center](https://csrc.nist.gov/glossary/term/requirement?utm_source=chatgpt.com)) |
| FR‑2 | Users shall be able to create a ride request specifying origin, destination, and preferred time, and then browse available ride offers that match those criteria. | Core ride‑sharing capability |
| FR‑3 | Drivers shall be able to publish ride offers and manage (accept/decline) incoming passenger requests. | Enables supply side of the platform |
| FR‑4 | The system shall integrate with the campus smart‑parking service to reserve an available parking bay for a driver who signals arrival. | Seamless campus experience ([FHWA Operations](https://ops.fhwa.dot.gov/publications/fhwahop18080/ch3.htm?utm_source=chatgpt.com)) |
| FR‑5 | The platform shall support in‑app payment so that ride costs (or parking fees) can be settled electronically between participants. | Removes cash handling friction |
| FR‑6 | Participants shall receive real‑time status updates (e.g., driver en‑route, pickup confirmed, parking reserved). | Minimises uncertainty |
| FR‑7 | Any rider or driver shall be able to trigger an SOS feature that shares live location with the University Safety Office and emergency contacts. | Safety requirement |
| FR‑8 | Administrative users shall be able to generate summary reports of ride counts, parking utilisation, and incident logs. | Supports operations governance |

## **3.2 Performance Requirements**

1. The application should open its core screens (login, ride list and offer‑posting) within ‘a few seconds’ on typical campus Wi‑Fi or 4 G connections, reflecting ISO 25010’s definition of performance efficiency without imposing hard millisecond limits. [ISO 25000](https://www.iso25000.com/index.php/en/iso-25000-standards/iso-25010?utm_source=chatgpt.com)[swavesglobal.com](https://swavesglobal.com/blog/top-5-practices-for-optimizing-mobile-app-loading-time?utm_source=chatgpt.com)
2. It should remain responsive when dozens of people use it at the same time during class change overs, a concurrency level drawn from mobile app performance best practice and easily scaled later through load testing. [Medium](https://medium.com/%40flutterwtf/mobile-app-performance-how-to-improve-and-test-it-0d4f0a9db3c7?utm_source=chatgpt.com)
3. Live parking status and trip updates ought to refresh often enough that users perceive the information as current seconds rather than minutes.
4. The service should be available throughout normal campus operating hours (roughly 06:00 – 23:00)

## **3.3 Usability Requirements**

1. New users should be able to ask for or offer a ride the first time they open the app, without any help from others.
2. Everyday tasks such as booking a ride should take only a few taps and very little typing. [Nielsen Norman Group](https://www.nngroup.com/articles/mobile-checkout-ux/?utm_source=chatgpt.com)
3. The app should spot and flag obvious mistakes in a form before you press “Submit,” so errors are avoided early [Nielsen Norman Group](https://www.nngroup.com/articles/slips/?utm_source=chatgpt.com)
4. Text and icons must keep enough contrast so everyone can read them easily.
5. Buttons and other touch areas need to be clearly designed so they’re easy to hit.
6. In user testing, the app should score around 70 out of 100 on the System Usability Scale, meaning most people find it ‘okay’ or better to use [measuringu.com](https://measuringu.com/interpret-sus-score/?utm_source=chatgpt.com)