

RMMM Table – Medication Tracker App

Risk ID	Risk Description	Probability	Impact	Risk Type	Mitigation Strategy	Monitoring & Management Plan
R1	Firebase Authentication failure (login/signup issues)	Medium	High	Technical	Use try-catch for auth calls, display clear error messages, fallback login retry	Monitor Firebase status, log auth errors, and use analytics to track auth success/failure rates
R2	App crashes or UI bugs during med addition or calendar view	Medium	Medium	Technical	Thorough unit & UI testing, input validation, error boundaries	Crashlytics (e.g., Firebase Crashlytics), regular QA testing
R3	Firebase quota or pricing limitations with free tier	Low	High	Technical /Operational	Monitor usage, optimize database reads/writes, enable Firestore indexing	Set alerts on Firebase console, upgrade tier if needed
R4	Data loss or sync failure between client and Firebase	Medium	High	Data Integrity	Use proper Firestore rules, offline persistence, and conflict resolution	Sync status indicators, retry failed writes, and backup data periodically
R5	Time zone-related issues for medication reminders	Medium	Medium	Functional	Use device time zone, store time zone with user profile, test reminders across zones	QA testing in multiple time zones, log reminder times
R6	Accessibility limitations for elderly or visually impaired users	Low	Medium	UX/Compliance	Follow accessibility best practices (font size, contrast, voiceover support)	Conduct accessibility audits, gather user feedback
R7	User confusion with action modal (taken/skipped interface)	Medium	Medium	UX	Use clear icons, onboarding tutorial, confirm before marking	Gather analytics on modal interactions, run user testing
R8	Privacy breach or weak Firebase rules allowing data leakage	Low	High	Security	Define strict Firestore rules, enable user-based read/write permissions	Regular rule audits, penetration testing
R9	History view shows incorrect or incomplete data	Medium	Medium	Data/Functional	Maintain normalized and timestamped structure in database	Test history calculations and rendering logic, use console logs