**Resources**

* Keep the **three tables** (Analysis/Design Tools, Prototyping Tools, Final Implementation Tools) but modify them to reflect your actual hybrid mobile prototype rather than the placeholder “C# Windows app.”
* Emphasise that **React Native + Expo Go + SQLite** were selected because they extend TM352 (mobile + data handling) and fit the academic goal of demonstrating a full-stack system.
* Add a **knowledge/skills subsection** where you explicitly tie module learning (TM352, TM354, M250) to practical application. E.g., TM354 informed your UML and requirements elicitation; TM352 informed your React Native prototype; M250 supported your OO analysis.
* Strengthen the **soft skills narrative**: show how communication and stakeholder consultation were limited but effective through prototypes and document reviews. Link this to T227 (change management and communication).

**2. Risks**

* Present a **formal risk register** with Likelihood, Impact, Mitigation, and Traffic Light status.
* Add “owner” or “responsibility” where relevant — even if this is “student” or “technical advisor.”
* Explicitly evaluate **which risks materialised** and how they affected planning. For example:
  + Limited stakeholder access (High risk, realised, mitigated via use of paper forms and prototyping).
  + Technology scope creep (High risk, realised, mitigated by adopting MVP strategy).
* This demonstrates critical reflection (LO4/LO5).

**3. Planning**

* Insert a short **time management reflection**:
  + Where progress matched your initial plan (requirements, modelling delivered on time).
  + Where it slipped (implementation longer due to learning curve in SQLite/React Native).
  + What was re-scoped or deferred (notifications, report exports).
* If possible, **include a Gantt chart or timeline diagram** in your appendix and refer to it here.
* Mention **MoSCoW prioritisation** (Must/Should/Could/Won’t) and how it helped manage time constraints.

**4. Lifecycle Choice**

* Your hybrid Waterfall + Iterative justification is strong. To make it excellent:
  + Link directly to **Sommerville (2016)** on lifecycle models.
  + Explicitly state that **Waterfall was used for early phases (requirements, domain modelling)** because of stable paper-form evidence and limited stakeholder access.
  + Iterative/Agile principles were then applied during prototyping and UI testing, to accommodate evolving insights.
* Add a line on why **full Agile** was not feasible: a solo developer and sparse stakeholder access.

**5. Critical Evaluation (to tick LO4/LO6)**

For each choice, add a short “why / so what” statement:

* Why React Native and SQLite instead of Xamarin or NoSQL?
* So what: ensured portability, offline-first capability, and compliance alignment.
* Why hybrid lifecycle?
* So what: balanced academic rigour with practical delivery constraints.