**Risk Assessment Report for Sprint 2 – *Bucket List Application***  
*Date: 11/05/2025*

# Introduction

Sprint 2 of the Bucket List mobile app project focuses on implementing the Bucket List Management and Task Generation (Suggestion) systems. As Business Analyst, this report identifies and analyzes potential risks that could affect Sprint 2, drawing on known issues from the bug report (e.g., incomplete README, missing .env.example, incorrect app name) and anticipated challenges in task/suggestion features. It offers a structured risk assessment with likelihood, impact, and priority ratings, followed by mitigation strategies. Effective risk management ensures the team can address hazards early, improving software quality and user experience.

# Risk Identification

We have identified the following key risks for Sprint 2, grouped by category:

* **Documentation and Setup Risks:** Incomplete or poor documentation (e.g. the **README** lacking details) can hamper development. Missing configuration examples (such as a missing .env.example file) can confuse developers during setup. These issues can lead to misunderstandings, delays, and errors.
* **Configuration Errors:** An incorrect app name or inconsistent identifiers (e.g. mismatched bundle IDs, package names, or UI labels) can cause build failures, store submission rejections, or branding confusion. Such misconfigurations can disrupt the release process.
* **API Reliability:** The Task Generation system may rely on external APIs (for example, for generating or fetching suggested tasks). API downtime or performance issues are critical risks: users are quick to abandon slow or inaccessible apps. Unreliable APIs can degrade the feature or break functionality.
* **Data Synchronization Issues:** If users can access their bucket lists on multiple devices or offline, data sync problems (delays, conflicts, lost updates) may occur. Unsynchronized data can frustrate users with inconsistent task lists.
* **Functional Bugs (Duplicate Tasks):** The Task Generation algorithm might create duplicate tasks. Duplicate data can lead to redundancy and user confusion. Without checks, the app’s list could fill with overlapping entries.
* **Suggestion Accuracy (User Trust):** Inaccurate or irrelevant task suggestions can erode user trust. Recommender systems that make wrong predictions risk losing user confidence. If suggestions feel off-target, users may disengage.
* **UI/UX Inconsistencies:** A disjointed interface (e.g. inconsistent design elements or navigation) can confuse users and harm brand perception. Without a unified design, the user experience suffers, potentially increasing support needs.
* **Scope and Requirements Clarity:** Unclear or changing feature requirements can introduce scope creep or overlooked tasks, risking incomplete delivery in the sprint.

# Risk Analysis

Each identified risk is analyzed for **Likelihood** (probability of occurrence) and **Impact** (severity on the project), yielding a **Priority** rating. The table below summarizes these ratings along with recommended mitigation actions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood | Impact | Priority | Recommended Actions |
| Incomplete or poor documentation (e.g. README) | Medium | High | High | Update and complete project docs; enforce documentation review checklist. |
| Missing .env.example configuration file | High | Medium | High | Provide a sample .env file; document required env vars; validate setup in onboarding. |
| Incorrect app name or identifiers | Low | Medium | Medium | Correct app naming in config files and UI; verify consistency before release. |
| External API downtime or performance issues | Medium | High | High | Use reliable API providers; implement caching/fallback; set up monitoring and alerts. |
| Data synchronization errors (cross-device) | Medium | High | High | Implement robust sync strategy (e.g. version control, conflict resolution, offline support); thorough sync testing. |
| Duplicate tasks generation | Medium | Medium | Medium | Add checks for existing tasks (unique IDs); refine generation logic to avoid overlaps. |
| Inaccurate or irrelevant task suggestions | Medium | Medium | Medium | Improve suggestion algorithms (use relevant data); allow user feedback on suggestions; perform accuracy testing. |
| UI/UX inconsistencies | Medium | Medium | Medium | Adopt a design system or style guide; conduct UI reviews and usability testing. |

# Risk Mitigation Strategies

For each risk, we recommend proactive mitigation:

* **Documentation and Setup:** Assign an owner to update the README and all key docs before or at the start of Sprint 2. Ensure the README clearly describes setup steps, dependencies, and feature descriptions. Add a sample .env.example with placeholders for all needed environment variables. Implement a peer review of documentation updates. Good documentation prevents wasted effort debugging ambiguous issues.
* **Configuration Errors:** Immediately correct the app name and identifiers in all relevant project files and metadata. Ensure version control history does not carry old names. Verify consistency by testing builds on target platforms. This prevents unexpected build failures or store submission problems.
* **API Reliability:** Choose stable, well-documented APIs for the suggestion system. As a safeguard, implement retry logic and local fallbacks (e.g. cached default tasks) when the API is unreachable. Monitor the API’s health and set alerts for downtime. Regular performance testing will reveal latency issues early. These steps minimize the chance that users experience downtime or slowness.
* **Data Synchronization:** Design a clear data model and use proven synchronization libraries or patterns. For offline use, implement local storage and syncing logic that handles conflicts (e.g. last-write-wins or user prompts). Test under network disruptions to ensure data integrity. Regular sync audits and logging help catch issues before they affect users.
* **Duplicate Tasks:** Modify the Task Generation logic to check for existing tasks before adding new ones (e.g. compare titles or use unique identifiers). Establish database constraints or filters to prevent duplicate entries. Include test cases for the duplication scenario. This will avoid redundant tasks that confuse users.
* **Suggestion Accuracy:** Tune the suggestion algorithm using relevant data (user preferences or categories) and guardrails to avoid irrelevant suggestions. Allow users to reject or rate suggestions, feeding back into the system. Conduct user testing of suggestions during development. Since incorrect recommendations can erode trust, continuous refinement based on feedback is key.
* **UI/UX Consistency:** Develop or reference a style guide (colors, fonts, layout patterns) and have a UI component library. All new screens should follow this guide. Conduct design reviews and walkthroughs with stakeholders. Prioritize consistency so users do not face a steep learning curve.
* **Scope Management:** Ensure requirements for both features are fully defined and agreed at Sprint 2 planning. Break features into well-defined user stories. Any change requests should be evaluated for impact on sprint goals. This avoids missing critical tasks due to ambiguity.

These mitigation strategies are drawn from best practices in risk response planning. Regular monitoring (daily stand-ups, sprint reviews) will help the team detect and respond to any of these risks as the work progresses.

# Conclusion

This risk assessment for Sprint 2 identifies several important hazards related to documentation, configuration, external dependencies, data integrity, functionality, and user experience. By analyzing their likelihood and impact, we prioritize addressing poor documentation, API reliability, and data sync issues as **high-priority** concerns. Implementing the recommended mitigation actions—such as completing project documentation, providing environment setup guides, enforcing UI standards, and building resilient API and sync mechanisms—will significantly reduce these risks. Ongoing risk monitoring and team communication will further ensure that Sprint 2 delivers the Bucket List features on time and with high quality.