

PROJECT DEVIATION REPORT

Deviation	Req ID/Source	SRS Baseline	Actual Implementation	Deviation Type	Rationale	Impact
Database Platform	Operating environment	“Mongo Atlas will be the database environment for this application	Supabase Postgres	Change	Easier for all group members to access the database since it’s a cloud interface.	<ul style="list-style-type: none"> • Schema redesign • Different SDKs
Authentication provider	Operating environment	“Authentication and authorization will be implemented through Auth0.”	Supabase Auth	Change	Consolidates platform and allows for custom registration/login/password reset pages.	<ul style="list-style-type: none"> • Different OAuth/JWT claims • Update middleware and tests
Transport security/tokens	R224	“Server implements JSON Web tokens for safe data transfer.”	Used framework-issued JWT via Supabase; no custom Token service	Change	Offload token insurance to managed provider	<ul style="list-style-type: none"> • Token validation differs from custom design • Update API guards
Image storage	R197	“System should be able to store images in the database.”	Store images (as URLs) in object (bucket) storage	Change	Better performance/cost	<ul style="list-style-type: none"> • Update upload limits • Update storage bucket
Grocery expiration tracking	Milestone 3 Acceptance	“Grocery expiration tracking functionality should be implemented.”	Not implemented.	Omission	Time/complexity; clarified as out-of-scope	
Admin activity log	R192	“All Admin activity should be recorded on a separate log.”	Basic audit only for recipe CRUD; no full admin log UI.	Partial (Omission)	Time/complexity; lower priority vs core user features.	<ul style="list-style-type: none"> • Reduced auditability
Newsletter subscription	R161	“Users should be able to subscribe to a newsletter...”	Not implemented.	Omission	Time/complexity ; No ESP integration yet.	<ul style="list-style-type: none"> • no email consents gathered. • Smaller engagement surface
UI overhaul	2.7 UI/UX Interface Mock-ups (UI01–UI23).	2.7 Interface mockups from page 17 to 46	Complete visual and layout redesign (navigation, cards, forms, color/typography), reflecting the new AI-driven search experience.	Change	Usability improvements; alignment with product vision; supports LLM search flows.	<ul style="list-style-type: none"> • Reimplemented new UI designs
Recipe Search via LLM natural language	UC10, R147, R152, R157, Milestone 3 acceptance	Search is filter-centric: “search for recipe based on filters such as cuisine, ingredient, popularity, dietary preferences, and reviews.	Natural-language (LLM) search that interprets prompts (e.g., “I want a salmon dinner	Change	Better discoverability; accommodates dietary constraints without manual	<ul style="list-style-type: none"> • Update UC10 flow, acceptance criteria, telemetry, and privacy notes

		Recipe page includes advanced search options and a search bar with recommendations/autofill; users can search from the Recipe page and apply filters.	without peanut”), retrieves and ranks best-match recipes from the DB.		filter; provides better user experience.	(prompt logging); define fallback behavior and quality metrics.
Recipe search via image upload (ingredient extraction)	UC10 – Search Recipe (User); Data Dictionary & Recipe Form image fields.	Search defined via filters (not images). SRS includes image fields (e.g., Recipe image URL/path; recipe form can upload an image) but not image-based search.	Users can upload a food photo; AI extracts ingredients and searches for matching recipes.	Addition	Multi-modal entry lowers friction; aligns with modern user expectations.	<ul style="list-style-type: none"> • Add upload endpoints, file limits/CDN policy, model governance, and accessibility notes; update security and PII sections.
Voice dictation for search	Recipe/Individual Recipe Page requirements (search bar).	“The recipe page should have a search bar.” (No voice input specified.)	Microphone button for speech-to-text in the search bar; hands-free queries.	Addition	Enhance accessibility and speed	<ul style="list-style-type: none"> • Update permissions flows • check browser compatibility • implement error handling for transcription • update non-functional requirements.
Recommendations engine: third-party API to in-house model	2.6 Operating Environment	“A third-party API [is] responsible for machine learning regarding recipe suggestions.”	Custom recommendation model trained and hosted by the team (no AWS/third-party service).	Change	Control over features and training data; lower vendor cost risk; easier to iterate.	<ul style="list-style-type: none"> • remove third-party dependency • add MLOps, monitoring, retraining cadence, and data privacy notes.