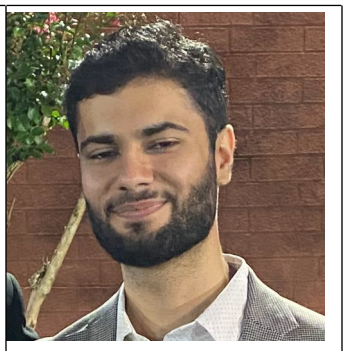
PROJECT PLAN

SP-05 — Grocery List App
CS 4850 - Section 03: Fall 2025
Aug 31, 2025

			
Drew Claerbout Documentation	Jose Garcia Documentation	Tyler Chetty Developer	Aman Bhayani Developer

Team Members:

Roles	Name	Role	Cell Phone / Alt Email
Documentation	Drew	Responsible for project reports, risk assessment, and SDLC documentation.	770-283-0379 prowlersdc@gmail.com
Documentation	Jose	Creates design diagrams, screen mockups, and compiles final report package.	678-761-4883 gjose8120@gmail.com
Developer	Tyler	Focus on feature integration (grocery list) and testing.	470.363.0302 tyler.chetty7@gmail.com
Developer	Aman	Develop basic app structure and UI. Set up user authentication.	404.884.4149 amanbhayani608@gmail.com
Project Owner or Advisor	Sharon Perry	Facilitate project progress; advise on project planning and management.	770.329.3895 Sperry46@kennesaw.edu

1.0 Project Overview / Abstract (Research)

Managing household groceries can be a challenge, especially with larger families. Current solutions (notes, texting) lack efficiency. Duplicate items and store-specific goods can lead to confusion. This project aims to build a dedicated mobile app to solve these coordination issues within a defined user group (a “household”).

Primary Objectives

1. Real-Time Synchronization: Provide a sub-second sync engine to ensure all users in a household view a consistent list state
2. Implement a structured data model for grocery items to reduce ambiguity, allowing for features like smart grouping by store
3. Define a clear set of operations with strategies for resolving conflicting concurrent edits
4. Design a backend that keeps household data secured, authenticates users, and manages permissions (admin vs. standard user roles within household)

Scope

We aim to create a mobile app using a cross-platform framework (Flutter). This app will allow user authentication and household management (creating a household, joining, leaving). The app will have the ability to categorize and sort items at user’s convenience. The app will update in real time, ensuring users are always seeing the most recent list.

The app will not have predictive item addition or any complex machine learning features. The app will not be able to process payment. It will also not have location based services beyond user-defined store preferences.

2.0 Project website

The URL for the website we will develop for the project is (may be subject to change): https://www.spgrocerylist.com

Deliverables - Specific To Your Project

Phase 1: Core Functionality (MVP) User Account System

– ability to create an account and securely log in. Shared Grocery List
– add members to a list, enabling collaborative editing. Real-Time Updates
– instant syncing of grocery list items across all members’ devices. CRUD Operations
– add, edit, delete grocery items with immediate reflection. Basic UI/UX Mockups
– each team member creates an individual design (Appendix C). Group Screen Mockups & Information Flow Diagrams
– included in the Software Design Document (SDD).

Phase 2: Expansion / Monetization

Third-Party Integration: integration with grocery store APIs for item prices and availability.
Monetization Features: ads, premium subscription, or shopping recommendations.
Scalability Improvements: performance optimizations to handle larger user groups.

Technical Deliverables

Framework Tutorials Completion: all members complete the Flutter tutorial (Appendix B).
Source Code Repository: hosted on a GitHub Organization account (not individual accounts).
Project Website: simple site containing project overview, documentation, and link to GitHub repo.
Unit and Integration Tests: validation of functionality and data synchronization.
Deployment Package: mobile build for both iOS and Android (demo-ready).

Documentation Deliverables

Software Requirements Specification (SRS): formal requirements documentation.
SoftwareDesign Document (SDD): includes:
Architecture diagrams
Group mockups
Data flow diagrams

Final Report (and Draft): includes tutorials (Appendix B), mockups (Appendix C), testing evidence, and results.
User Manual / Quick Start Guide: for non-technical users to understand app functionality.
Presentation Deliverables
Final Presentation Video: overview of the project, demonstration of core features, and reflection on challenges/solutions.

Group Meeting Schedule Date/Time

The group meeting will be scheduled on Wednesdays every week at 8 PM.

Collaboration and Communication Plan

The group will meet synchronously (online) using Discord from wherever it is convenient.

Project Schedule and Task Planning

Project Name: SP-05 Red Grocery List
Report Date: 8/31/2025

					Milestone #1				Milestone #2				Milestone #3				C-Day	
Phase	**Tasks**	**Complete%**	**Current Status Memo**	**Assigned To**	**09/02**	**09/09**	**09/16**	**09/23**	**09/30**	**10/07**	**10/14**	**10/21**	**10/28**	**11/04**	**11/11**	**11/18**	**####**	**11/04**
Requirements	Conduct Studies	50%		Jose	20	12												
	Define requirements	100%		Drew		15												
	Review requirements with SH	0%		Aman		10	15											
	Get sign off on requirements	0%		Tyler			5	10										
Project design	Define tech required *	0%		Drew				10	4									
	Database design	0%		Aman				5	10	10								
	Front End design	0%		Tyler						10	10							
	Develop working prototype	0%		Jose, Tyler						10	10							
	Test prototype	0%		Drew, Jose							5	10	5					
Development	Review prototype design	0%		Aman, Tyler								8	5	10				
	Rework requirements	0%		Jose								8	10	20	20			
	Document updated design	0%		Drew											10	10		
	Test product	0%		Jose, Drew										8	5	20		
Final report	Presentation preparation	0%		Drew												15	10	10
	Poster preparation	0%		Jose														10
	Final report submission to D2L and project owner	0%		Drew														5
			Total work hours	**375**	**20**	**37**	**25**	**25**	**14**	**30**	**25**	**26**	**20**	**38**	**35**	**45**	**10**	**25**

* formally define how you will develop this project including source code management

Legend	
Planned	
Delayed	
Number	Work: man hours

Progress Planning and Tracking

Other Plans: Like Risk Assessment (if applicable)

Data Security & Privacy

Risk: Unauthorized access to user accounts or data leakage.
Impact: High: loss of user trust, potential data breach.
Mitigation: Enforce secure authentication, and proper database rules.

Scope Creep / Missed Deadlines

Risk: Spending too much time adding Phase 2 features (monetization, store integration) before MVP is complete.
Impact: High: final deliverables may not be ready on time.
Mitigation: Prioritize Phase 1 (MVP) and only expand if time allows.

Team Readiness (Learning Curve with Flutter)

Risk: Not all members complete the required Flutter training/tutorial.
Impact: High: uneven skill levels could slow development.
Mitigation: Require tutorial completion as proof (Appendix B) before coding begins.

Version Control Plan

We will utilize Git as our version control system, hosted on a centralized platform (GitHub). The workflow will be based on the Feature Branch Workflow, which is well-suited for collaborative development and continuous integration. The main branch will always reflect a stable, production-ready state. All new work, whether it's a feature, bug fix, or chore, will be performed in a dedicated, short-lived branch created from the latest main branch. This isolation ensures that the main branch remains stable and allows for parallel development without conflicts.

Each feature branch will be named descriptively using a convention like feature/[ticket-number]-[short-description] (e.g., feature/GA-12-add-auth-ui). Development will be committed in small, logical units with clear and concise commit messages. To integrate changes, developers will open a Pull Request (PR). Each PR must pass all automated CI checks (linting, testing, builds) and require at least one approved code review from a peer before it can be squashed or rebased and merged back into the main branch. This process enforces code quality, facilitates knowledge sharing, and maintains a clean, linear project history.