"THE GENESIS"

"Power to new beginnings"

Phase 1: UI Prototyping & Workflow Documentation

Project Phase: 1 – UI Prototype, Flow Diagram & Structural Planning

Project Title: Spaza Shop Regulatory & Management System (SSRMS)

Date: 18/04/2025

Overview

Phase 1 serves as the foundational planning and prototyping stage of the SSRMS project. During this phase, Team Genesis will conceptualize and design the User Interface (UI) through simple wireframes and UI flow documentation. This process ensures all stakeholders have a shared visual understanding of the system before development begins.

Objectives of Phase 1

- Design a basic but clear representation of the system interface using wireframes.

- Illustrate navigation flow between key pages.

- Store all visual assets and related documentation in the docs/wireframes/ directory.

- Commit all materials to the project's GitHub repository for team accessibility and version control.

Summarized Required Deliverables

The Genesis must:

1. Design the User Interface (UI) using tools such as Figma, Balsamiq, or hand-drawn sketches.

2. Save Wireframe Images in .png or .jpg format and upload them.

3. Create a Wireframe Description Document (wireframe_description.md) describing the UI flow.

Example Folder Structure

```
├── docs/
│   └── wireframes/
│       ├── home_page.png
│       ├── business_listings.png
│       └── wireframe_description.md
```

Note: This structure that Sir expectants and will check when marking phase 1 therefore it must be maintained.

Project Manager's Workflow Instructions

-To ensure clear communication and progressive delivery, all team members must adhere to the following structure and sequence: Before the System designer, back-end and front-end start with each responsibility they must first have small communications/meetings in person or virtually so, whereby they explore ideas and choose which UI Prototype, Flow Diagram & Structural Planning will best suit the Spaza Shop Regulatory & Management System (SSRMS).

1. System Designer Responsibilities

- Lead the UI and user flow design using approved tools.

- Ensure each key page is represented in a wireframe.

- Describe the navigation between pages.

- Upload all designs to doc's/wireframes/.

- Submit the wireframe_description.md document.

- Once complete, formally notify the Front-End Developer for review.

2. Front-End Developer Responsibilities

- Discuss with back end the languages and tools suited for the "SSRMS".

- Review the wireframes and confirm UI feasibility.

- Take note of components, layout, and screen transitions.

- Begin preparing front-end implementation outlines (in the next phase).

- Communicate required adjustments to the System Designer if needed.

- Once reviewed, notify the Back-End Developer.

3. Back-End Developer Responsibilities

- Study the wireframe documentation to identify back-end data needs that can be done for phase 1, creating also a clear path for "what can be done after this "e.g. phase 2.

- Map out database entities, APIs, and integration points (can also include for the for upcoming phases as well).

- Provide feedback on any inconsistencies between the UI flow and expected data operations.

Teamwide Rules and Standards

- Sequential Handover: Work must follow the order — System Designer → Front-End Developer → Back-End Developer.

- Approval Before Progression: No team member may begin their task without confirmation that the previous step is complete.

- Version Control: All files (images and markdown) must be pushed to GitHub& the genesis group on WhatsApp if needed using proper commit messages.

- No Folder Alteration: The directory structure provided is standard and must not be changed.

Take note that if they can create their own useful strategies for team collaboration then they can follow those steps but making sure to keep the folder structure wanted by the Lecturer

After the last the Responsibility performed by Back-end before submission Genesis's **Tester will verify all the documents and steps performed**

Submission Checklist

Deliverable
Home Page Wireframe
Business Listings Wireframe
Wireframe Description Document
Full Upload to GitHub

Closing Remarks

This phase is not about perfection but clarity. Wireframes should be simple, intuitive, and accurately represent the planned user experience. Every team member is responsible for ensuring that the project progresses smoothly from design to implementation, with feedback and coordination built into each handoff.

This rubric is the same as what the Lecturer will use as it provides a structured assessment for Phase 1 deliverables uploaded to GitHub. The evaluation is based on clarity, completeness, documentation, collaboration, and adherence to instructions.

Category	Score (0-2)	Score (3-4)	Score (5-6)	Score (7-8)	Score (9-10)	Max Score
GitHub Repository Setup & Structure	No repository created OR repository lacks essential folders/files	Repository exists but missing key files like README, docs, wireframes	Repository structured but lacks some documentation or issues with access	Proper structure, good documentation, and team members added	Repository is well-organized, fully documented, and follows all requirements	**10**
Product Vision Document	Missing or lacks clarity	Basic content but lacks key details	Contains most elements but lacks depth or consistency	Well-defined, clear structure, and good reasoning	Comprehensive, well-structured, strong rationale for project vision	**10**
Wireframes & UI Prototypes	No wireframes uploaded	Only rough sketches with no explanation	Basic wireframes uploaded but missing descriptions	Good wireframes with clear UI flow	Detailed, professional wireframes with well-explained UI logic	**10**
Software Product Management Plan	No management plan uploaded	Basic document but lacks key elements like roles and milestones	Covers some elements but lacks structure or reasoning	Well-organized, clear roles, milestones, and risk management	Highly detailed, well-structured plan covering all project management aspects	**10**
Meeting Attendance & Logs	No meeting logs uploaded	Few meetings recorded, poor documentation	Meeting logs exist but lack details on attendance or discussions	Well-documented meetings with clear notes and attendance	Detailed meeting logs, well-structured discussions, and full participation	**10**
Team Collaboration & Participation	No collaboration observed, only one member contributing	Few commits from multiple team members	Some members contributing, but commits are not balanced	All members contributing with regular commits	Active collaboration, clear version control, and pull requests	**10**