

Name: *Arul Prasanth K*

Title: *BinGo(wastage collections)*

Roll No: *2023239001*

Branch: *Computer Science*

BinGo: SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) REPORT

1. Introduction to SDLC

The **Software Development Life Cycle (SDLC)** is a systematic process used to design, develop, test, deploy, and maintain software systems. It provides a structured framework that ensures the development of high-quality software within a planned schedule and budget. SDLC helps in effective requirement analysis, better project planning, risk reduction, and improved software quality.

For the development of **BinGo: A Real-Time Waste Collection and Recycling Tracking Mobile Application**, SDLC plays a vital role in managing mobile application development, backend services, real-time tracking, and user-centric design within a fixed academic timeline.

2. Need for an Appropriate SDLC Model

Selecting an appropriate SDLC model is essential for project success. The choice depends on project size, complexity, requirement flexibility, available resources, and time constraints. Since **BinGo** is an **individual mini project**, involving UI/UX design, mobile application development, and real-time features, a flexible and iterative approach is required.

3. Evaluation of Different SDLC Models

3.1 Waterfall Model

The Waterfall model follows a linear and sequential approach where each phase must be completed before the next begins.

Limitations for BinGo:

- No flexibility for requirement or UI changes

- Late testing increases risk
- Not suitable for mobile applications with evolving design

Conclusion:

✗ Not suitable for BinGo.

3.2 Incremental Model

The Incremental model develops the system in multiple increments.

Limitations for BinGo:

- Requires early finalized architecture
- Integration complexity for individual development
- Limited support for continuous UI/UX refinement

Conclusion:

✗ Not ideal for BinGo.

3.3 Prototype Model

The Prototype model focuses on early UI prototypes.

Limitations for BinGo:

- Suitable only for requirement validation
- Not effective for backend-heavy features
- Leads to rework if prototypes change significantly

Conclusion:

✗ Insufficient as a full SDLC model.

3.4 Spiral Model

The Spiral model emphasizes risk analysis and iteration.

Limitations for BinGo:

- Designed for large, high-risk projects
- Excessive documentation and analysis

- Not suitable for an academic mini project

Conclusion:

✗ Overly complex for BinGo.

4. Selected SDLC Model: Agile Development Model

4.1 Overview of Agile Model

The **Agile Development Model** is an iterative and incremental approach where software is developed in small cycles called **sprints**. Each sprint includes planning, development, testing, and review, allowing continuous improvement and flexibility.

4.2 Why Agile is Chosen for BinGo

The Agile model is selected for **BinGo** due to the following reasons:

- Supports iterative development and flexibility
- Allows UI/UX improvements throughout development
- Enables early testing and bug detection
- Suitable for mobile application development
- Ideal for individual projects
- Ensures timely completion within academic constraints

Hence, Agile is the most suitable SDLC model for this project.

5. SDLC Phases Applied to BinGo

5.1 Requirement Analysis

Includes problem definition, objectives identification, requirement gathering, and preparation of the **Software Requirements Specification (SRS)**.

5.2 System Design

Includes preparation of:

- Use Case Diagram

- ER Diagram
- Class Diagram
- Sequence Diagram
- System Architecture Diagram

5.3 Implementation

Backend development using **FastAPI (Python)**, mobile application development using **React Native (Expo)**, and database management using **PostgreSQL**.

5.4 Testing

Functional testing, integration testing, and validation of real-time tracking features.

5.5 Deployment

Preparation of the system for execution and demonstration.

5.6 Maintenance

Bug fixing, performance optimization, and future enhancements.

6. User Stories for BinGo Application

6.1 User (Citizen)

- As a user, I want to register and log in securely.
- As a user, I want to request waste pickup using my location.
- As a user, I want to select waste type for proper disposal.
- As a user, I want to track the waste collection vehicle in real time.
- As a user, I want to view my pickup history.

6.2 Waste Collection Worker

- As a worker, I want to view assigned pickup tasks.
- As a worker, I want to see pickup locations on a map.
- As a worker, I want to update pickup status.
- As a worker, I want to share live location during collection.

6.3 Administrator

- As an admin, I want to view all pickup requests.
 - As an admin, I want to assign workers to requests.
 - As an admin, I want to monitor worker performance.
 - As an admin, I want to view analytics and reports.
-

7. Agile Sprint Schedule

Sprint	Phase	Duration	Period
Sprint 1	Requirement Analysis & Planning	2 weeks	11-12-2025 to 24-12-2025
Sprint 2	System Design	3 weeks	25-12-2025 to 14-01-2026
Sprint 3	UI/UX Design (Figma)	2 weeks	15-01-2026 to 28-01-2026
Sprint 4	Backend Development	4 weeks	29-01-2026 to 25-02-2026
Sprint 5	Mobile App Development	4 weeks	26-02-2026 to 25-03-2026
Sprint 6	Integration, Testing & Review Prep	2 weeks + 1 day	26-03-2026 to 06-04-2026

8. Conclusion

The **Agile Development Model** is the most suitable SDLC model for the **BinGo: Real-Time Waste Collection and Recycling Tracking Mobile Application**. Its iterative nature, flexibility, and sprint-based execution ensure efficient development, continuous testing, and timely completion before the academic review. Compared to traditional SDLC models, Agile provides better adaptability, improved quality, and effective time management, making it ideal for this individual mini project.