# Design Document: Bank Simulation

## 1. Introduction

The Bank Simulation program simulates a simplified bank service system where customers arrive, wait in line if necessary, and are served by a teller. This document provides an overview of the design and implementation details of the simulation.

## 2. Purpose

The purpose of this simulation is to model the operation of a bank with the following objectives:

* Simulate customer arrivals and departures.
* Track waiting times for customers.
* Calculate statistics such as total customers processed and average waiting time.

## 3. System Overview

### Components:

* **Event Class**: Represents an arrival or departure event with associated time and transaction length.
* **SL\_PriorityQueue Class**: Priority queue implementation to manage events based on their time.
* **ListQueue Class**: Queue implementation to manage customers waiting in line.
* **simulate() Function**: Main function to initiate and control the simulation.
* **processArrival() Function**: Handles logic for processing arrival events.
* **processDeparture() Function**: Handles logic for processing departure events.

### Flow:

1. **Initialization**:
   * Initialize queues and variables.
   * Read input from file containing arrival and transaction times.
2. **Simulation Loop**:
   * Process events from the priority queue (eventListPQueue).
   * Handle arrival events by either serving the customer immediately or adding them to the queue.
   * Handle departure events by either serving the next customer in line or marking the teller as available.
3. **Output**:
   * Print messages for each event processed.
   * Calculate and print final statistics after all events are processed.

## 4. Detailed Design

### Event Class

* Attributes:
  + type: Type of event ('A' for arrival, 'D' for departure).
  + time: Time of the event.
  + length: Transaction length for the event (only applicable to arrivals).

### SL\_PriorityQueue Class

* Implementation of a sorted priority queue using a linked list.
* Operations:
  + add(): Add an event to the priority queue.
  + remove(): Remove an event from the priority queue.
  + peek(): View the next event without removing it.
  + isEmpty(): Check if the priority queue is empty.

### ListQueue Class

* Implementation of a standard queue using a linked list.
* Operations:
  + enqueue(): Add an event to the end of the queue.
  + dequeue(): Remove and return the event at the front of the queue.
  + peekFront(): View the event at the front of the queue without removing it.
  + isEmpty(): Check if the queue is empty.

### simulate() Function

* Initiates the bank simulation.
* Reads input from file and initializes queues.
* Manages the simulation loop until all events are processed.
* Updates current time and processes events based on their type.

### processArrival() Function

* Processes an arrival event:
  + Adds the customer to the teller if available.
  + Otherwise, adds the customer to the bank queue.

### processDeparture() Function

* Processes a departure event:
  + Removes the served customer from the bank queue.
  + Updates waiting time statistics if applicable.
  + Schedules the next departure event or marks the teller as available.

## 5. Assumptions

* **Assumptions**:
  + The input file (events.txt) is correctly formatted and contains valid data.
  + Customers are served in the order they arrive.