Touch Test write up

# Assumptions

The client and IFA are expected to define the split between pots. If money is deposited into a pot, resulting in it exceeding the allowance, a GIA will automatically be used. If one does not exist, it will be created.

Design Decisions

* Idempotency middleware used to prevent requests being received multiple times
* The software uses pennies instead of pounds for the following reasons:
  1. Precision and Accuracy
     1. Working with pennies allows for a higher level of precision in financial calculations. Since pennies represent smaller units than pounds, it reduces the risk of rounding errors that may occur when dealing with floating-point numbers.
  2. Consistency in Data Storage
     1. Storing monetary values as integers (pennies) ensures consistency in the data structure. It simplifies data storage and retrieval processes, reducing the likelihood of discrepancies caused by different data representations.
* Use of Decimal Library to Prevent Floating-Point Issues
  1. To mitigate potential precision and rounding issues associated with floating-point arithmetic, a dedicated decimal library is to be used.
* Performing the allocation process within a single database transaction (ACID)
  1. Atomicity.
  2. Data Consistency.
  3. Isolation.
  4. Durability.
  5. Rollback on Failure.
  6. Ensuring Business Logic Completeness.
* Dependency injection
  1. Loosely coupled program.
     1. Modularity.
     2. Ease of Maintenance.
     3. Parallel Development.
     4. Flexibility and Adaptability.
  2. Allowed the mocking of services while testing.
* GORM as the ORM
  1. The decision to use GORM as the Object-Relational Mapping (ORM) tool was based on several factors. GORM, being a popular and well-maintained ORM for Go, offers a high level of abstraction for database interactions. Its expressive syntax and built-in support for features like eager loading, associations, and migrations streamline the development process. Gorm resolver supports read and write replicas that can be used in the future to increase performance.
* Go Struct validation
  1. The Go lang Validator library was chosen to facilitate input validation within the application. This library offers a straightforward way to define and enforce validation rules on data structures, ensuring that input adheres to the specified criteria.