1. Entwurf
   1. Server

Wir haben unseren Server in folgende Einheiten aufgeteilt: **QueueManager**, **ClientManager** und **Sender** sowie 2 **Queues** zur Speicherung von Nachrichten.

#TODO Abarbeiten von dropmessage, getmsgid und getmessages beschreiben.

**QueueManager** ist verantwortlich für Speicherung von Nachrichten in die Holdbackqueue und Übertragung dieser Nachrichten in die DeliveryQueue.

Jede erhaltene Nachricht kommt zuerst in die Holdbackqueue. Nach dem Speichern der Nachricht in Holdbackqueue prüft er die Größe der Holdbackqueue, ob sie von der Größe nicht die Hälfte von DeliveryQueue erreicht.

* Wenn das der Fall ist, bedeutet es für den QueueManager, dass in der Holdbackqueue eine Lücke entstanden ist. Dann füllt er diese Lücke mit einer Fehlernachricht und schiebt sie in die DeliveryQueue mit der größten Nachrichtennummer in der Lücke.
* Im anderen Fall tut er keine weiteren Aktivitäten.

**Sender** dient dazu, Nachrichten an den Lese-Client aus der DeliveryQueue zu senden und Informationen über den Lese-Client zu aktualisieren.

Er fragt die letzte Nachrichtennummer, die dem Client gesendet wurde, beim ClientManager ab.

Dabei überprüft der Sender, ob der Client dort schon existiert. In diesem Fall prüft er den gespeicherten Zeitstempel und setzt die letzte Nachrichtennummer des Clients zurück, wenn die vergangene Zeit vom Zeitstempel bis zur aktuellen Zeit ein in Konfiguration vorgegebenes Zeitintervall überschreitet. Danach wird der Zeitstempel des Clients aktualisiert und die Nachrichtennummer auf die kleinste Nachrichtennummer in der DeliveryQueue gesetzt.

Dann holt er sich die Nachrichten aus der DeliveryQueue, wählt die benötigte Nachricht aus und sendet sie an den Client.

**ClientManager** verwaltet eine Liste von registrierten Lese-Clients mit zugehörigen Informationen(ClientPid, letzte gesendete Nachrichtennummer und Zeitstempel von letzter Kommunikation mit dem Client).

Ein Client landet in dieser Liste, wenn er Nachrichten vom Server anfordert. Bevor er dort gespeichert wird, prüft der Sender, der die oben genannte Aktion initiiert, ob der Client dort schon existiert.

**HoldbackQueue** und **DeliveryQueue** haben eine gemeinsame generische Implementierung. Wir realisieren die Queues mit einfachen Lists, da sie für Speicherung von geordneten Mengen ausreichen, und lassen sich einfach mit Funktionen aus der Standard Library zu verwalten. Sie unterstützen folgende Aktionen:

* Einfügen einer Nachricht ans Ende der Queue
* Ankleben von mehreren Nachrichten ans Ende der Queue
* Holen aller Nachrichten aus der Queue als eine Liste.