

Hold It System Design Document

Team: RKJ

Written By:

Kegan, Ryan, Joseph

Table of Contents

1. Introduction
 - 1.1. Purpose of the System
 - 1.2. Design Goals
 - 1.3. Definitions, Acronyms, and Abbreviations
2. Current Software Architecture
3. Proposed Software Architecture
 - 3.1. Overview
 - 3.2. Subsystem Decomposition
 - 3.3. Access Control and Security
 - 3.4. Global Software Control
 - 3.5. Boundary Conditions
4. Subsystem Services
 - 4.1. Glossary

1 Introduction

1.1 Purpose of the System

The Purpose of the Hold it application is to create an automated system in which spots in lines can be held in exchange for monetary compensation. Customers will be able to either purchase a spot in a line as well as wait in line at the behest of someone else. This application will help to eliminate what many consider to be the worst part of events like purchasing tickets, getting a new phone, or even getting a spot for black friday.

1.2 Design Goals

Usability: Hold It should be easy to use for smartphone users.

Cost: The Hold It application should be made with minimal expenses.

Performance: Hold It should perform well on windows devices.

Rapid Development: Hold It will be made before the end of spring term.

Scalability: Hold It should be able to take on a large amount of users.

1.3 Definitions, Acronyms, and Abbreviations

Holdee: A client who is holding a spot in line.

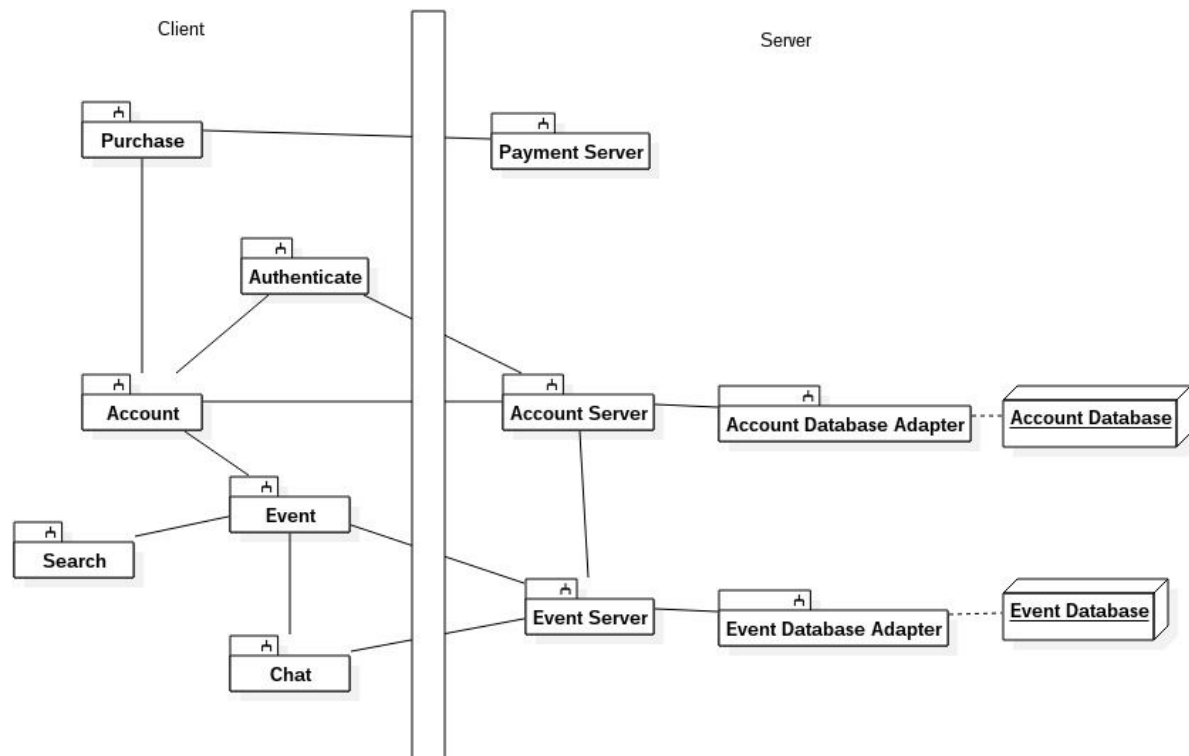
Customer: A client who is purchasing a spot in line

Spot: An offer to hold a spot at an event.

Tag: Label attached to an event to help with identification.

3. Proposed Software Architecture

3.1 Overview



3.2 Subsystem Decomposition

Beginning with the databases and servers:

Payment Server: The purpose of the payment server is to send the payment data off to a third party to be held and transferred at completion of a transaction.

Account Database: The account database holds all account information and uses that information to verify logins as well as edit account information.

Event Database: The Event database holds Events and chats for a limited amount of time. Events are created and Holdees can create spots inside of them, when a spot is chosen by a customer a chat is created until the transaction is finished.

Next the client subsystems:

Purchase: The Purchase subsystem creates payments when a spot is selected. It passes on payment holds to the server and database as well as cancels payments when necessary.

Authenticate: The authentication subsystem works closely with the account subsystem to verify users when logging in, and handling any issues that occur during the login process. It also logs out users.

Account: This subsystem allows for account creation and account editing. It also allows for Customers to become Holdees and users to report one another.

Search: The Search subsystem allows users to search for events in a variety of different ways. It also uses the user's location to find nearby events.

Event: This is the largest subsystem as it contains both events and spots. New events can be created and edited here, spots can be added by users and removed by customers selecting them.

Chat: This subsystem is closely created to an event as chats are created when a spot is chosen by a customer. Users can send messages, get notified of messages and check the location of the person they are chatting with.

3.3 Access Control and Security

Passwords and other important information such as payment info will be encrypted and held in the account database. The payment database will be securely encrypted and hold actual payments until the process is fully verified by both parties.

Admins will have the power to ban users and cancel payments but not the power to manipulate accounts and payments. Normal users will only have access to their own information and will not be able to view other users profiles past the rating they see when selecting a spot.

3.4 Global Software Control

The event subsystem takes requests to manipulate events and save data by passing it through the event server to the event database. The chat subsystem is directly created by the event subsystem whenever a spot is chosen, the chat subsystem can then access and store information directly through the event server. The search system interacts with event by asking it for a list of events and using the location provided by the gps to sort through these events.

The account subsystem manipulates accounts through the account server and database. For the account subsystem to even be touched (besides when creating an account) a client has to go through the authentication subsystem first and verify account

information with the account database. Authenticate and account directly interact with each other to ensure security of accounts.

The payment subsystem sends payment information securely and directly through the payment server to the payment database. An account will choose a spot in the event subsystem and be given a price. The payment information will then be retrieved from the account database and passed back to the account and on to the payment subsystem.

3.5 Boundary Conditions

Upon startup the application will go directly into the authentication subsystem and remain until a new account is either created or verified. The application will return to this state when logout is selected.

Upon unexpected termination of the application the search subsystem and chat subsystem will keep the user's location in the same place, with the chat subsystem displaying to any other users that the client is currently offline. All spots selected by the client will remain selected until the timer on said spots runs out. Any held purchases, at the end of the timer, will be processed if it was the customer whose application was terminated and canceled if it was the holder's application. Unselected spots being held by holders will enter an idle mode where they are not able to be selected but are still shown on the event page. Events will not be affected by application termination. As long as the application remains open the authentication subsystem will not be affected.

