

DESIGN DOCUMENT

Hubba - Account System

Version 1.3

Prepared By: Development Hell

Class: CECS 491B-05

Date: April 16, 2023

Github Repository:

<https://github.com/DevelopmentHellaHell/SeniorProject>

Team Leader

Kevin Dinh

Members

Garrett Tsumaki

Bryan Tran

Jett Sonoda

Tien Nguyen

Darius Koroni

POC: kevin.dinh@csulb.edu

Revision History

Version	Overview	Date
1.0	Start of Design Document	3/20/2023
1.1	Finished High Level Design	3/22/2023
1.2	Finished Low Level Design	4/8/2023
1.3	Adjusted Design based on WireFrame	4/15/2023

Table of Contents

Table of Contents	2
Overview	3
Requirements to Satisfy	4
High-Level Design	9
Low Level Design	16
Change Password	16
Get Account Settings	21
Change Account Settings (Email)	24
Get Rental History	27
Search Rental History	28
Booking Cancellation	30
Relational Table(s)	32

Overview

This document provides all the information needed for the design of the Account System. Included in this document are both the High and Low level designs for understanding the general system, the flow of how the feature operates, and any potential improvements in debugging and scalability.

Requirements to Satisfy

Reset Password	
Precondition	User completed Email Verification
Acceptance	<ul style="list-style-type: none">• Anonymous users• User must be on Registration/Login view
Requirement input	A new valid password can be set
Success criteria	All of the following conditions must be fulfilled: <ul style="list-style-type: none">• User can log in with the newly set password• User data is updated in the database
Failure criteria	Any one of the following conditions occurring constitutes failure: <ul style="list-style-type: none">• Password can't be reset at the moment• User can log in with previous passwords• Password stored is not identical to input password
Failure response	Prompt the user "Password reset failed. Refresh page or try again later."
Non-functional requirement	Database query results within 5 seconds
Priority	High

Settings (Account Modification)	
Precondition	User selects "Settings"
Acceptance	<ul style="list-style-type: none">• Authenticated users• User must be on Account view

Requirement input	<p>User input confirming action</p> <p>Any one of the following conditions must be fulfilled:</p> <p>User can change/update their credential/contact information such as:</p> <ul style="list-style-type: none"> - First Name - Last Name - Email (new email will need to be verified again using OTP) - Password
Success criteria	<p>All of the following conditions must be fulfilled:</p> <ul style="list-style-type: none"> • Account view reflects the change(s) • User data is updated in database
Failure criteria	<p>Any one of the following conditions occurring constitutes failure:</p> <ul style="list-style-type: none"> • Account can't be updated at the moment • Account does not reflect the change(s)
Failure response	Prompt the user "Account settings failed. Refresh page or try again later."
Non-functional requirement	<ul style="list-style-type: none"> • Database query results within 5 seconds • User must confirm new settings from options of yes or no <ul style="list-style-type: none"> ◦ Selection of no returns the user to their previous view • A change in password must be confirmed with the correct input of the old password
Priority	Medium

Rental History View	
Precondition	User selects "View History"
Acceptance	<ul style="list-style-type: none"> • Authenticated users • User must be on Account view
Requirement input	N/A

Success criteria	<ul style="list-style-type: none"> - If user has no booking, view is empty - Otherwise, rental history is shown in chronological order of bookings
Failure criteria	<p>Any one of the following conditions occurring constitutes failure:</p> <ul style="list-style-type: none"> • User has a rental history but can't see any of them • Rental history is not in chronological order • Rental history is not identical to actual history of bookings
Failure response	Prompt the user "Rental history failure to load error. Refresh page or try again later."
Non-functional requirement	Database query results within 5 seconds
Priority	High

Change Rental History View to Calendar Format	
Precondition	<p>All of the following conditions must be fulfilled:</p> <ul style="list-style-type: none"> • User selects "View History" • User selects calendar view formatting
Acceptance	<ul style="list-style-type: none"> • Authenticated users • User must be on Account view
Requirement input	N/A
Success criteria	<ul style="list-style-type: none"> - If user has no booking, view is empty - Otherwise, rental history is shown in chronological order of bookings in a calendar format
Failure criteria	<p>Any one of the following conditions occurring constitutes failure:</p> <ul style="list-style-type: none"> • User has a rental history but can't see any of them • Rental history is not in chronological order • Rental history is not identical to actual history of bookings • User does not see a calendar view of their rental history

Failure response	Prompt the user “Rental history calendar formatting error. Refresh page or try again later.”
Non-functional requirement	<ul style="list-style-type: none"> • Database query results within 5 seconds • Calendar business rules follow all calendar-specific rules listed under “Scheduling System” with the following exceptions: <ul style="list-style-type: none"> ◦ Calendar view starts with the month the user account was created and ends with the current month • Calendar functionality follows all calendar-specific features listed under “Scheduling System” with the following exceptions: <ul style="list-style-type: none"> ◦ Filtered listings used under the filter features return the user’s rental history listings and not available listings ◦ User cannot book a listing from this view
Priority	Low

Search in Rental History	
Precondition	<p>All of the following conditions must be fulfilled:</p> <ul style="list-style-type: none"> • User must select “View History” • Rental history must not be in calendar format
Acceptance	<ul style="list-style-type: none"> • Authenticated users • User must be on Account view
Requirement input	User input confirming action on Rental History view to search <ul style="list-style-type: none"> - User can sort the results - User can filter the results
Success criteria	The related booking will be displayed
Failure criteria	Matching booking is not displayed
Failure response	Prompt the user “Rental history search error. Refresh page or try again later.”

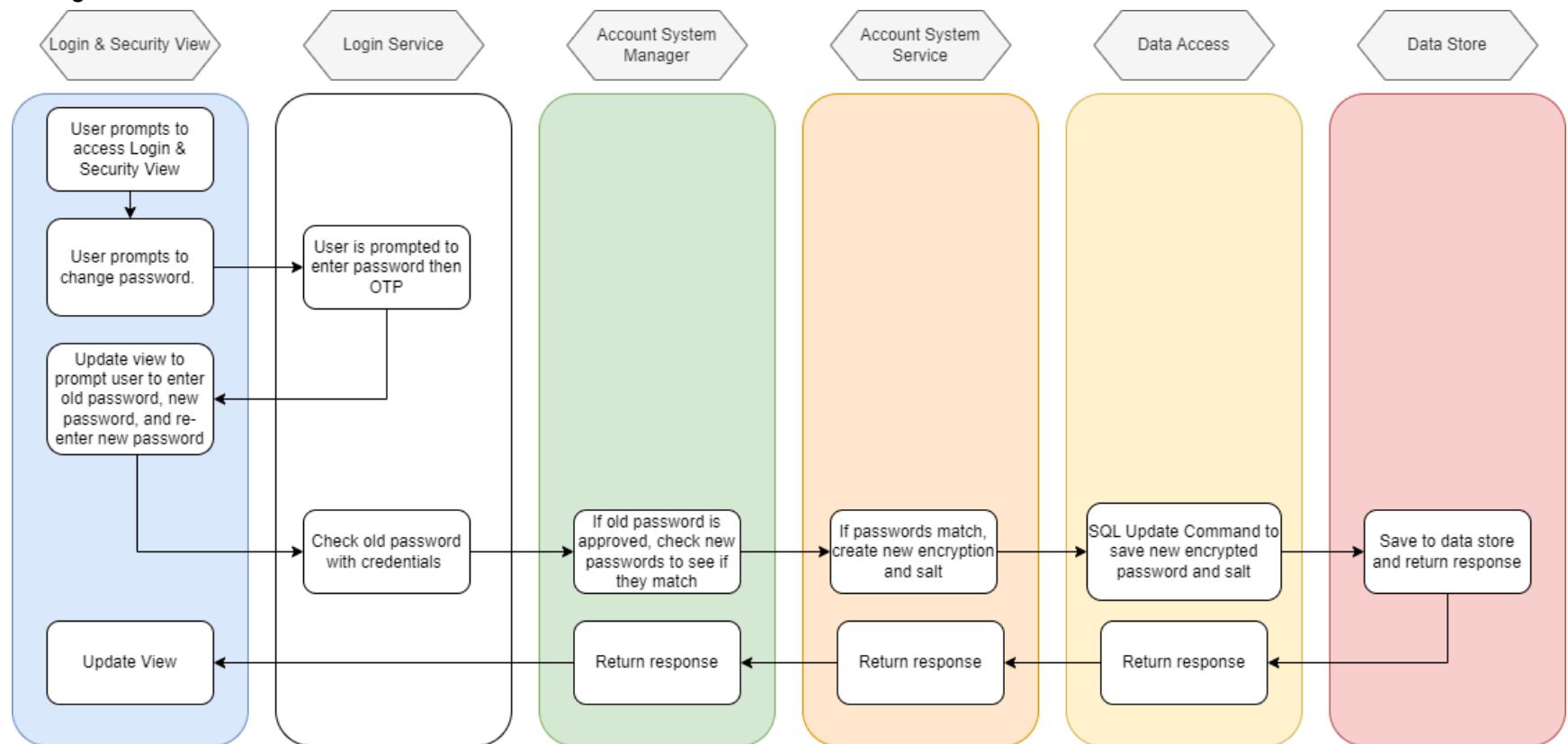
Non-functional requirement	Database query results within 5 seconds
Priority	Low

Rental Cancellation	
Precondition	<p>All of the following conditions must be fulfilled:</p> <ul style="list-style-type: none"> • User must have an active booking • User must select “View History”
Acceptance	<ul style="list-style-type: none"> • Authenticated users • User must be on Account view
Requirement input	<p>All of the following conditions must be fulfilled:</p> <ul style="list-style-type: none"> • User chooses to cancel a booking • User input confirming action for cancellation
Success criteria	<p>All of the following conditions must be fulfilled:</p> <ul style="list-style-type: none"> • There is a confirmation box of cancellation the selected booking • There is communication to the host of the cancellation • A confirmation email is sent to the user with cancellation information including date, location, and time frame • Booking is updated in database
Failure criteria	<p>Any one of the following conditions occurring constitutes failure:</p> <ul style="list-style-type: none"> • Confirmation of booking cancellation is not sent to user • Confirmation of booking cancellation is not sent to host • Removal of booking in database is incomplete • Message on application view does not indicate cancellation
Failure response	<p>Determine whether a booking cancellation was sent to the database</p> <ul style="list-style-type: none"> - If confirmed, notify user confirming booking cancellation “Booking cancellation confirmed” - If unconfirmed, retry once. On repeated failure, notify user “Booking cancellation not confirmed. Refresh page or try again later.”

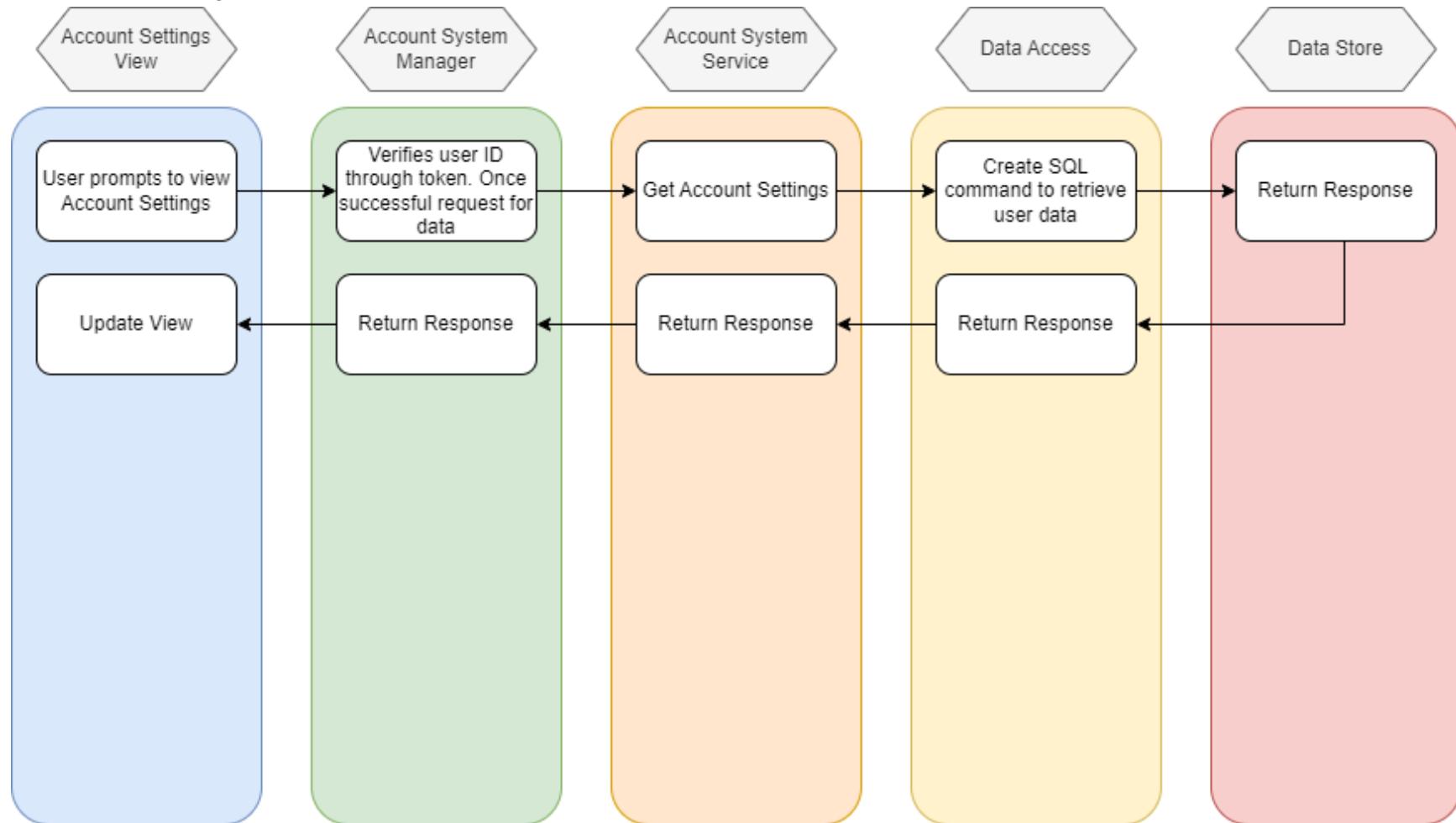
Non-functional requirement	<ul style="list-style-type: none">• Update database in less than 5 seconds• User must confirm rental cancellation from options of yes or no• Selection of no returns the user to their previous view
Priority	Critical

High-Level Design

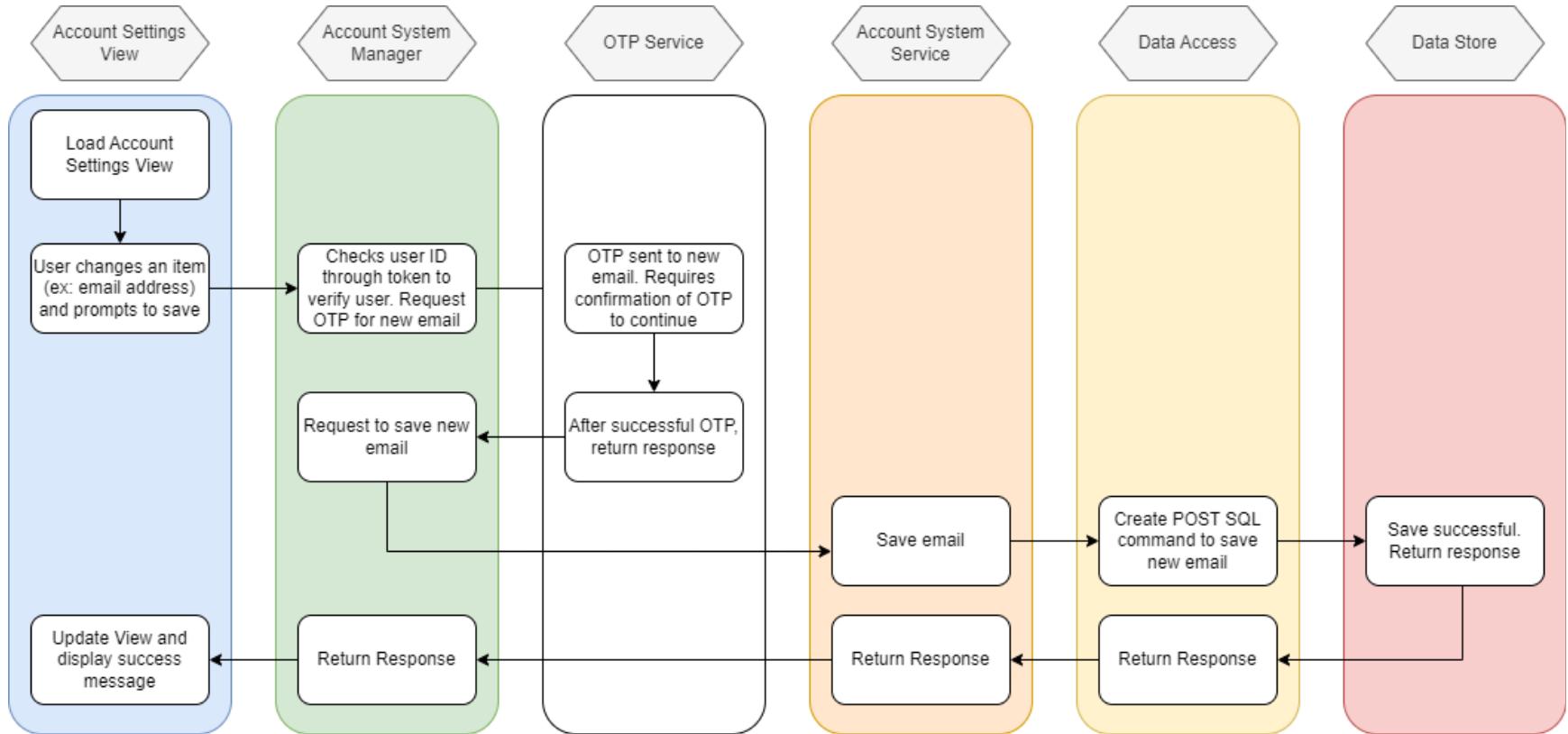
Change Password



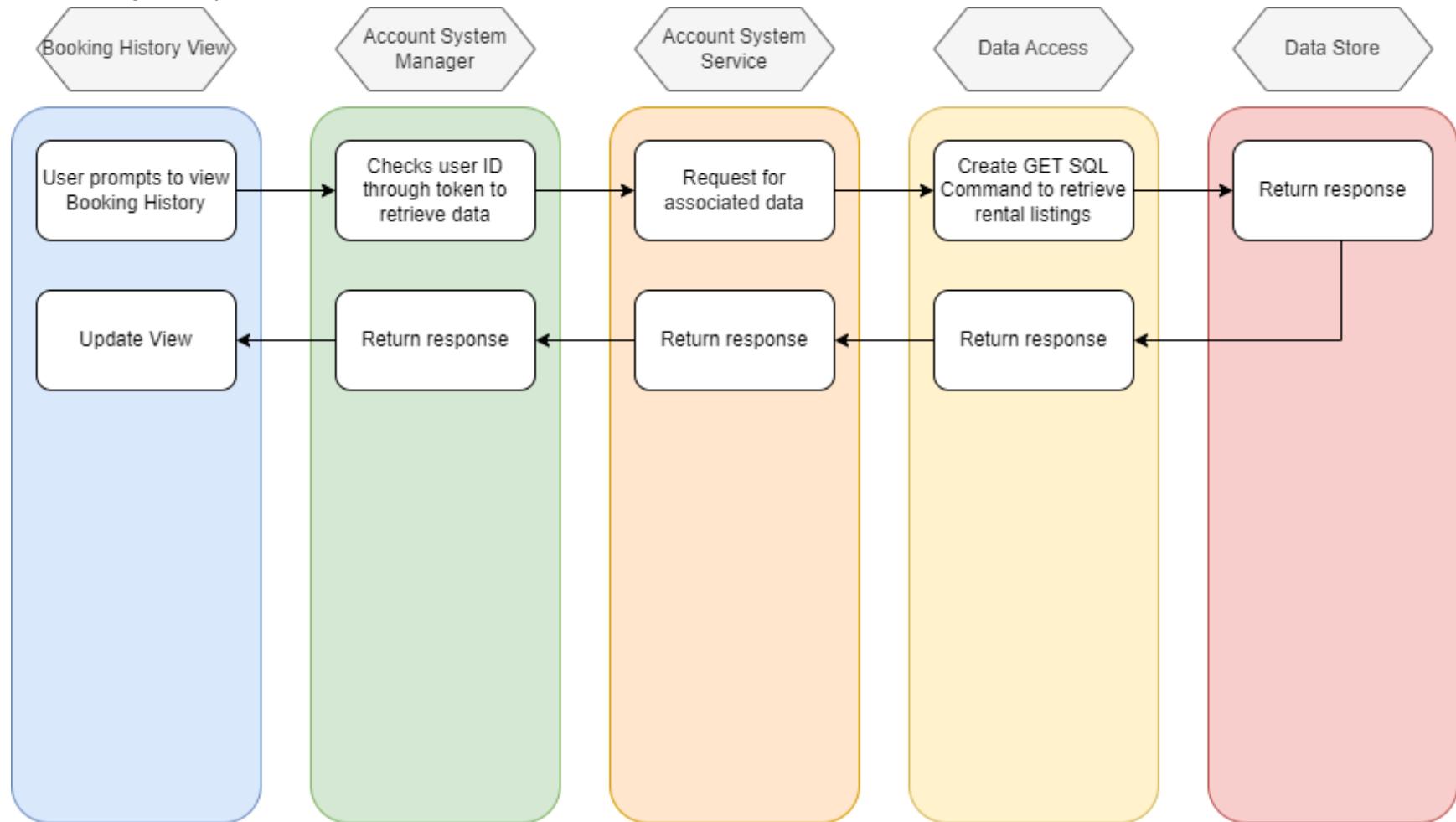
Get Account Settings



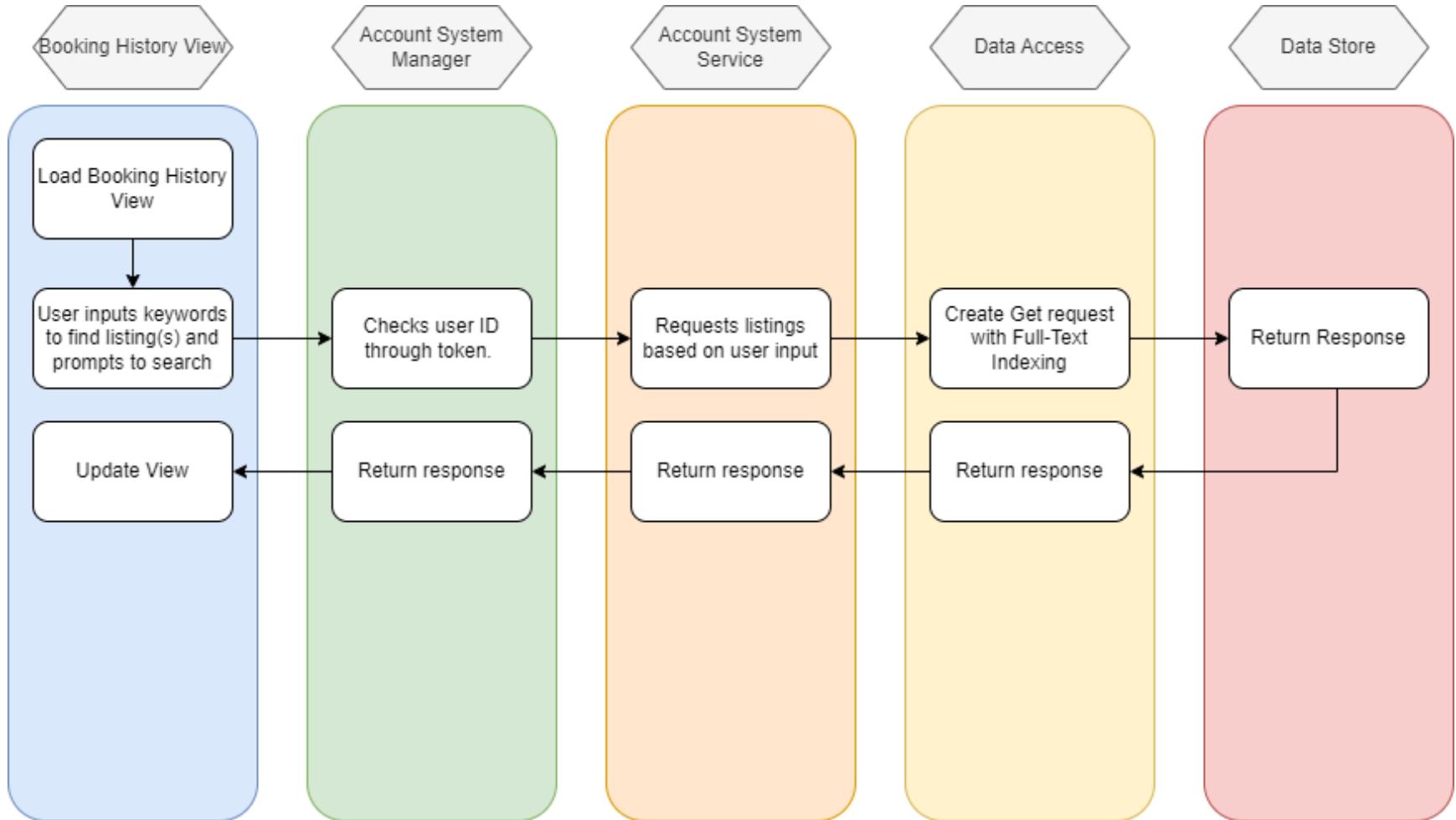
Change Account Settings (Email)



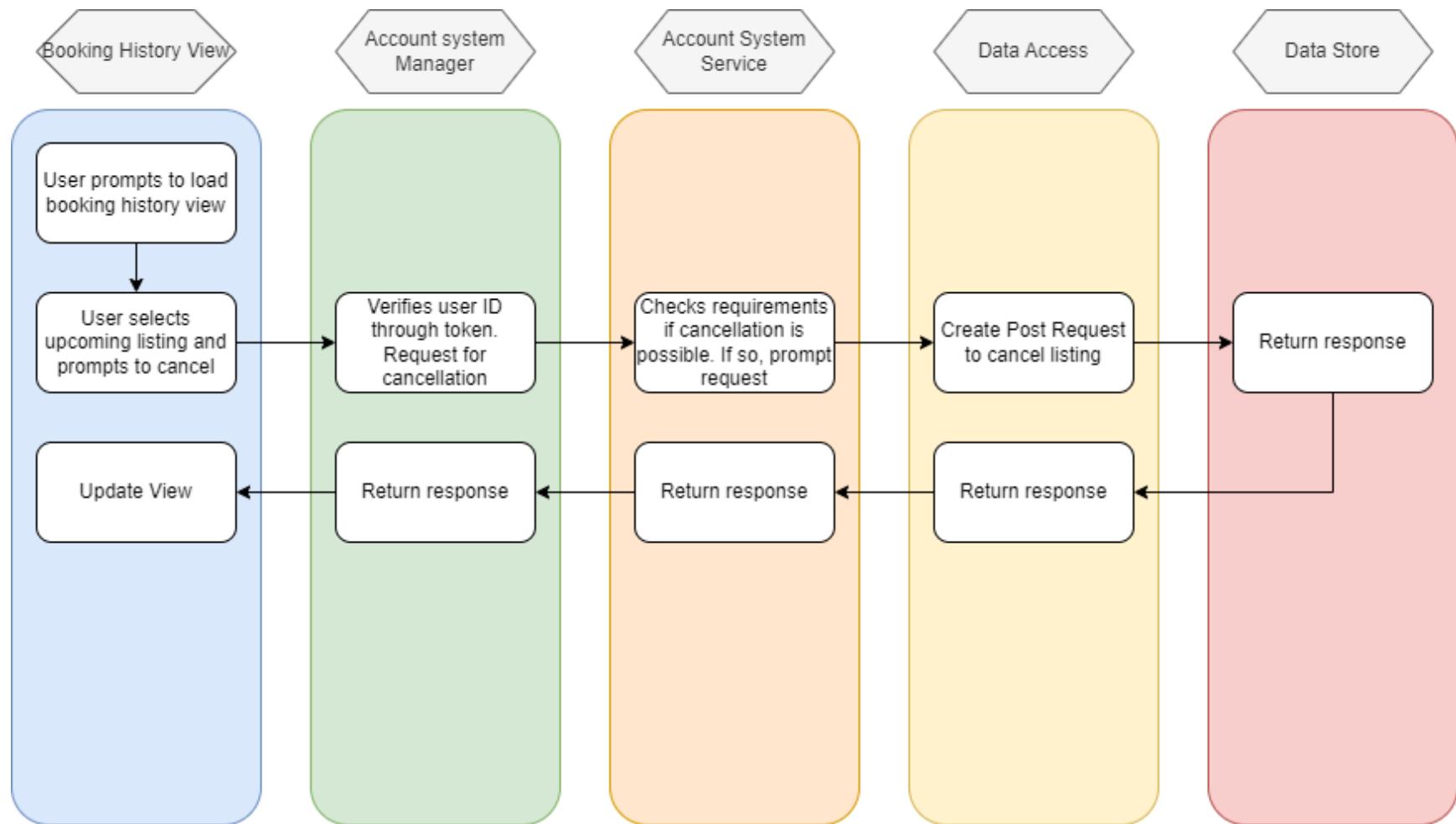
Get Booking History



Search in Rental History



Rental Cancellation

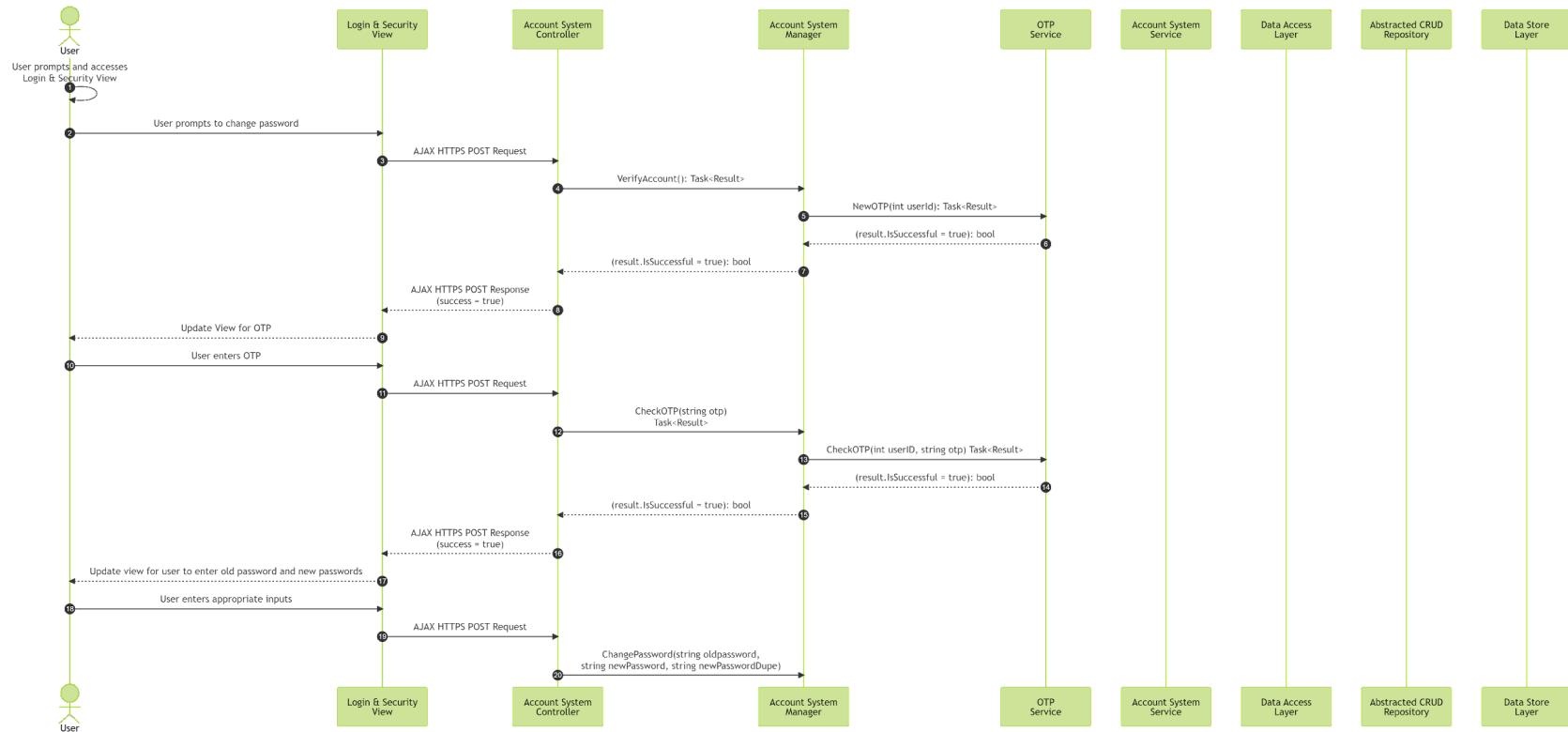


Low Level Design

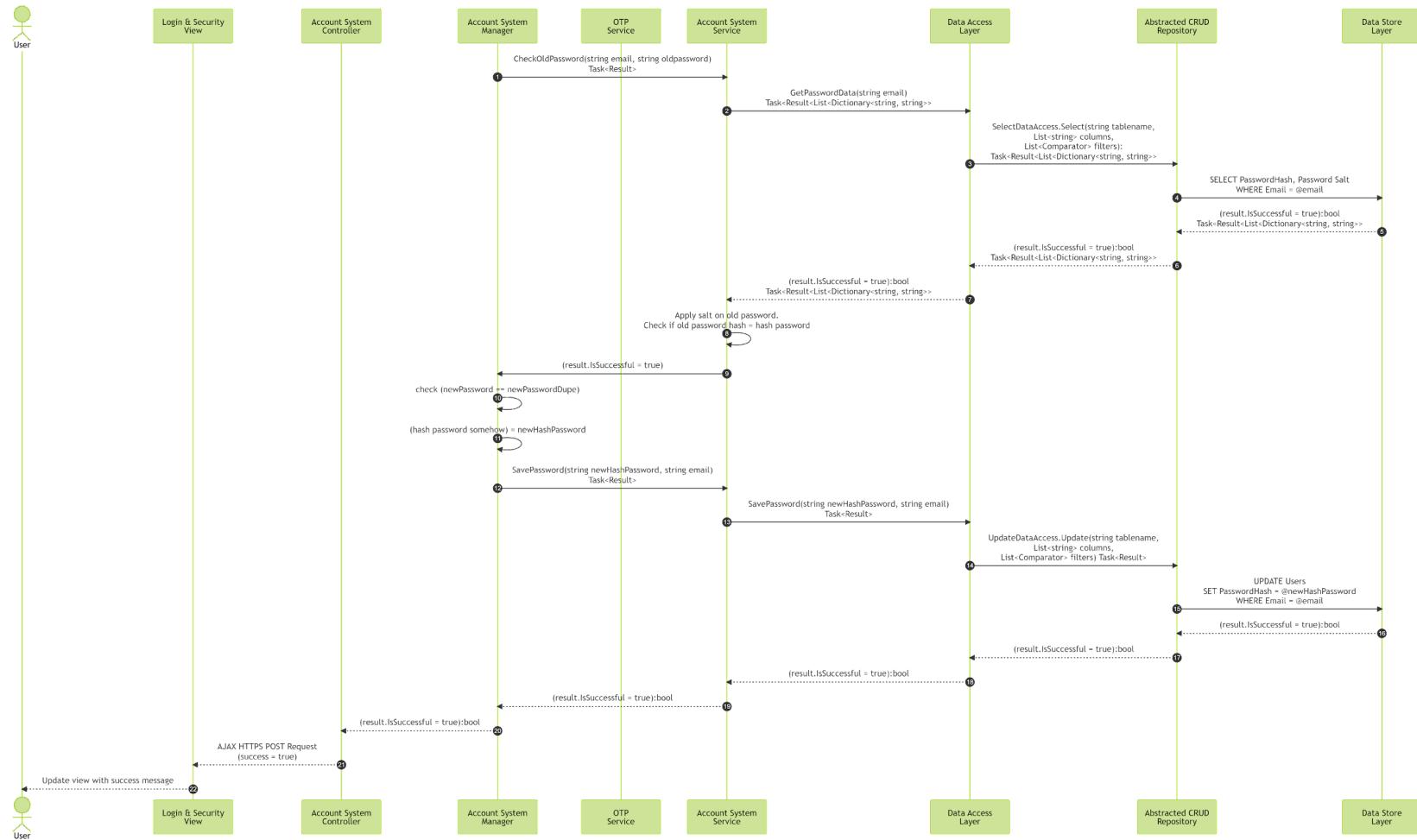
Change Password

Success Case: User enters correct credentials and changes password

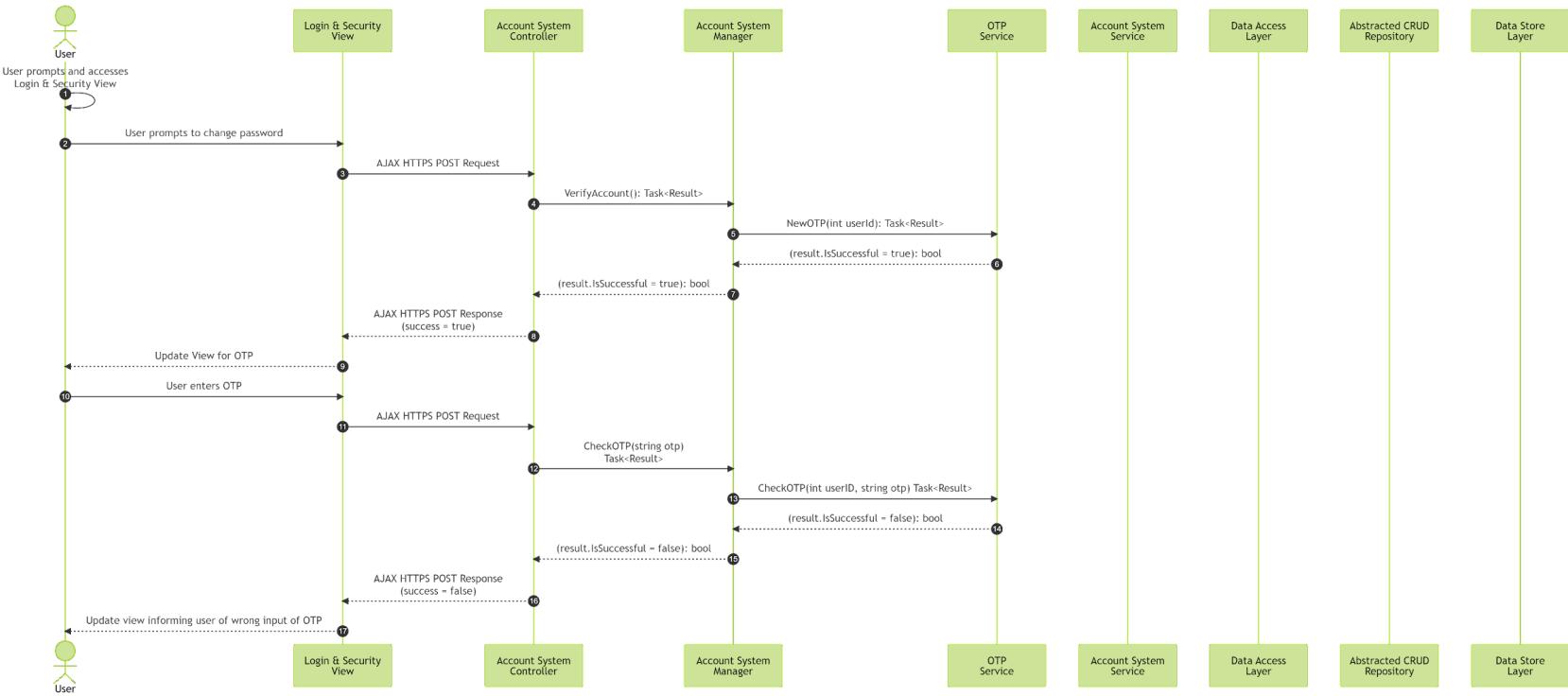
Pt.1



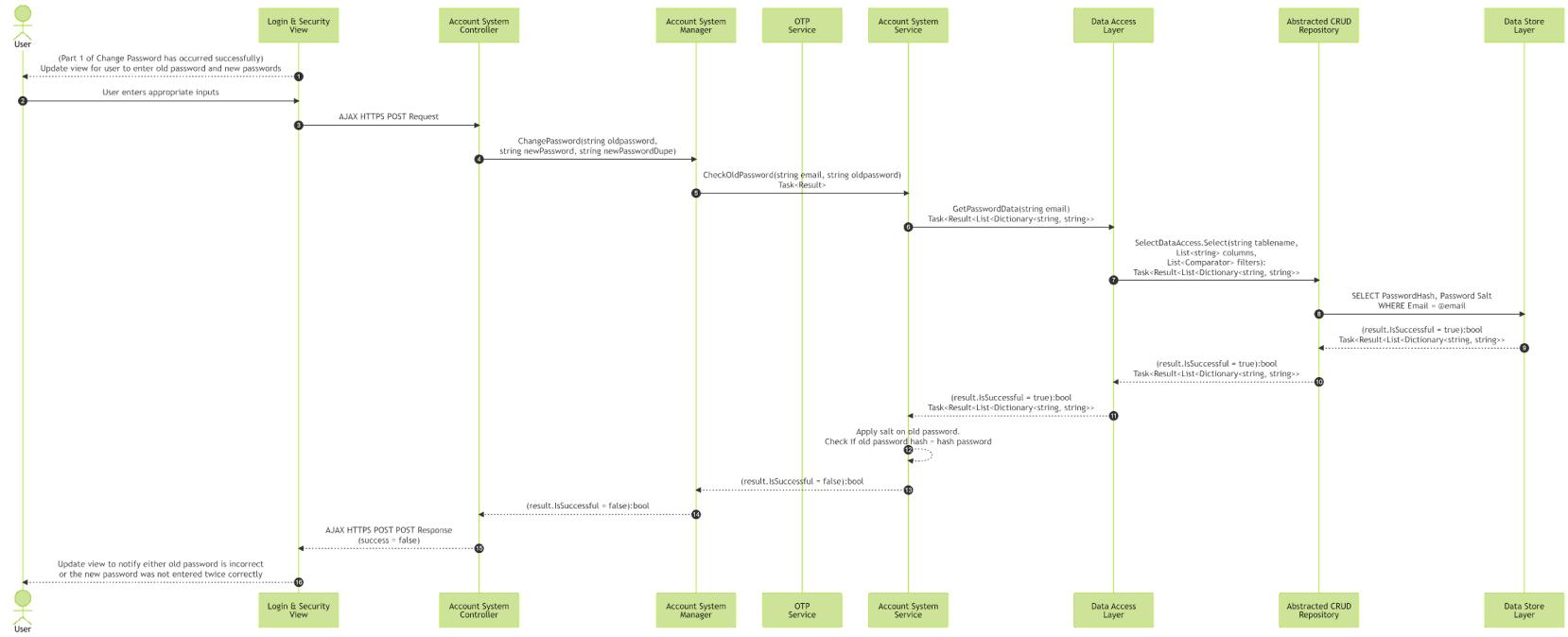
Pt.2



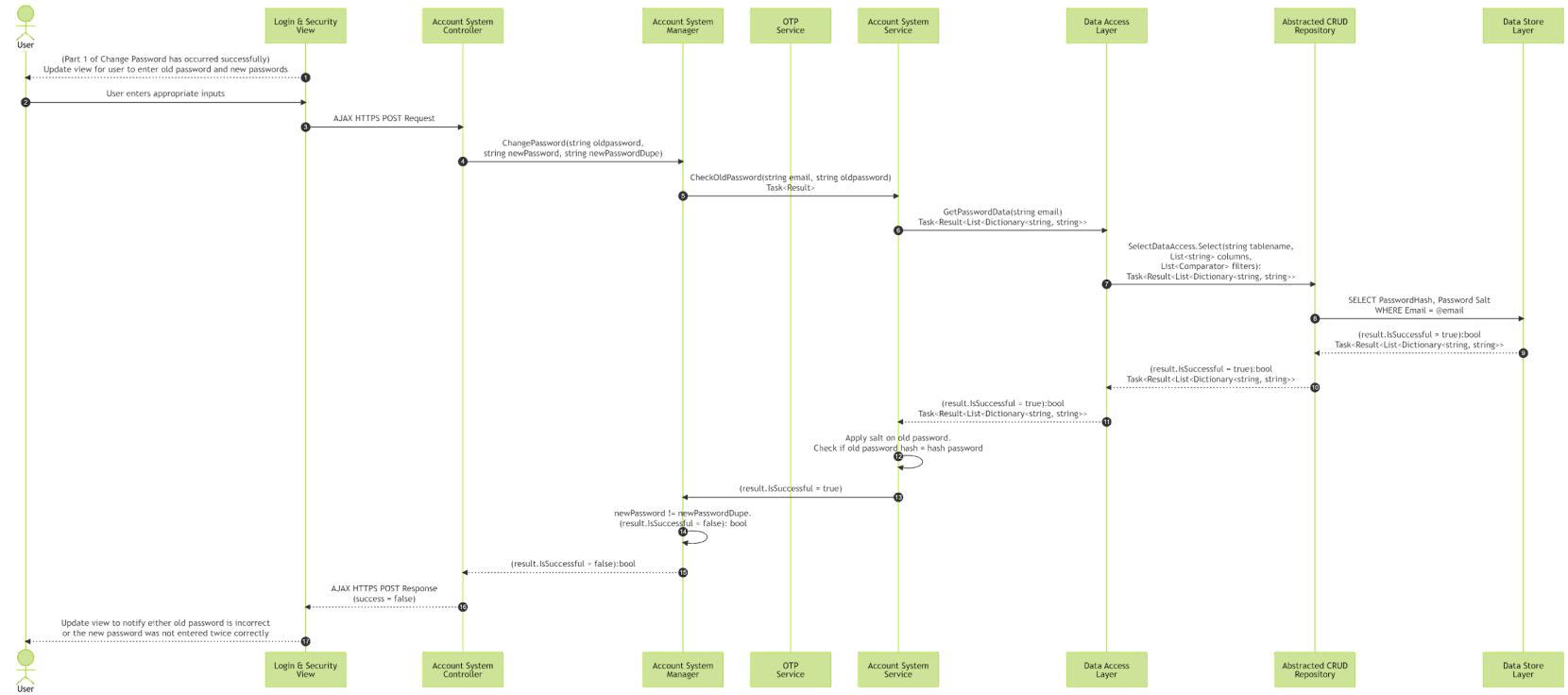
Failure Case: User fails OTP



Failure Case: User does not enter correct original password

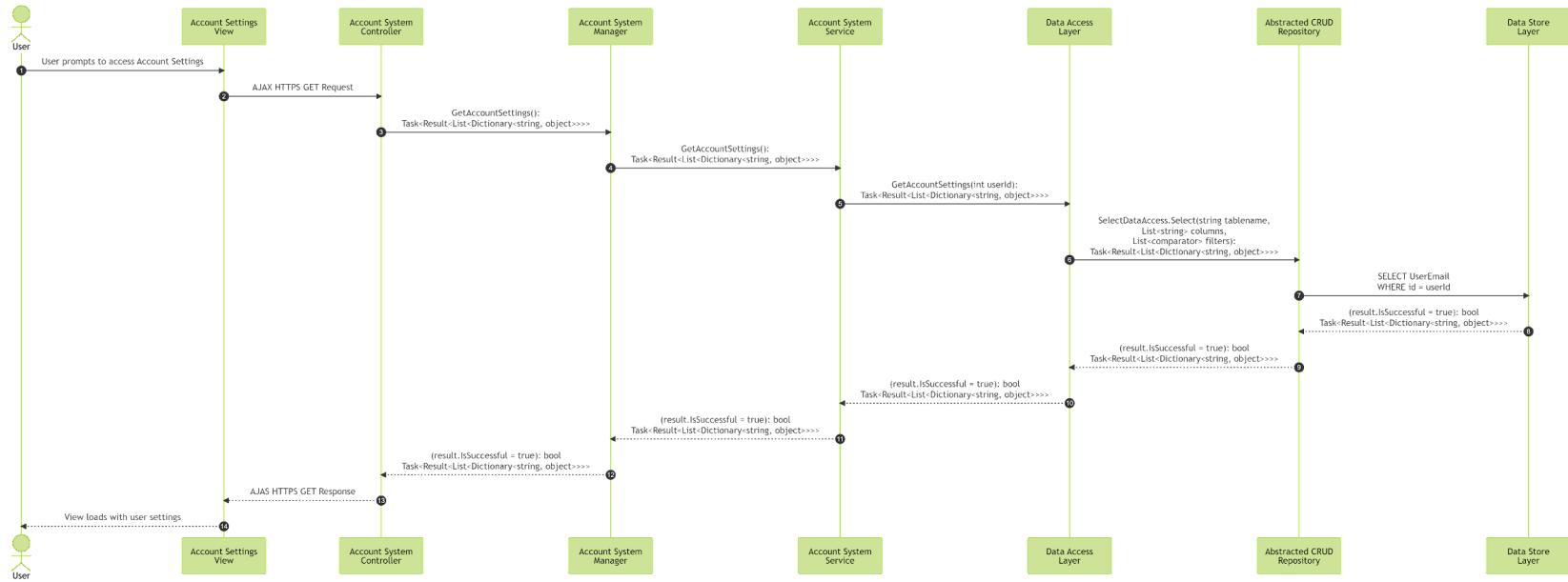


Failure Case: User's new passwords do not match

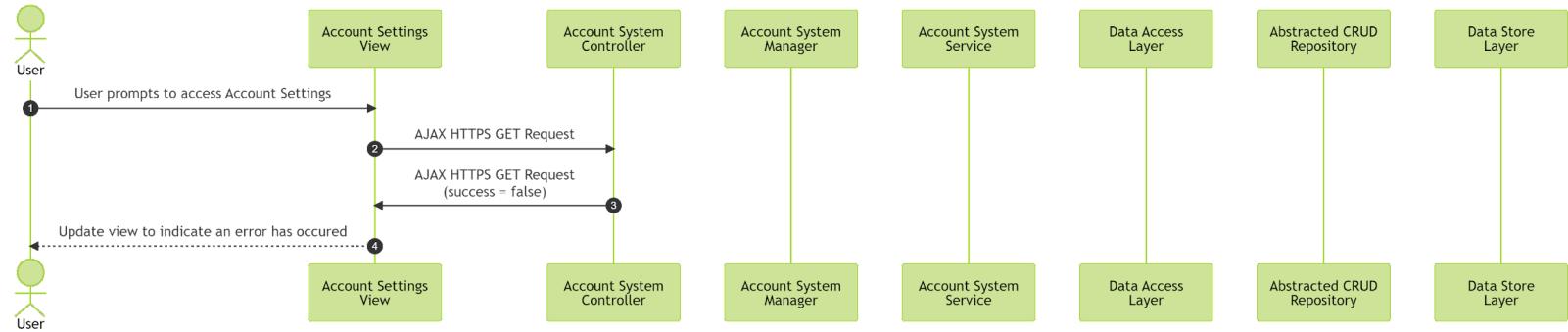


Get Account Settings

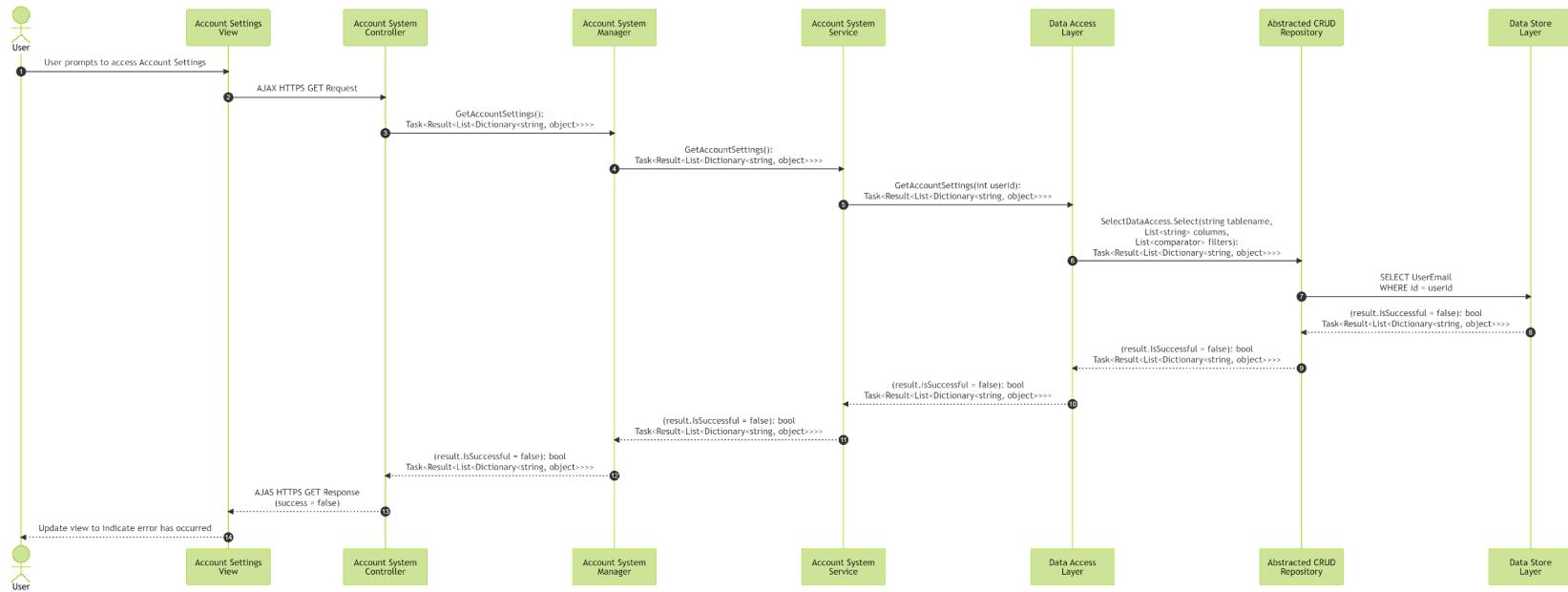
Success Case: User is able to navigate to Account Settings View



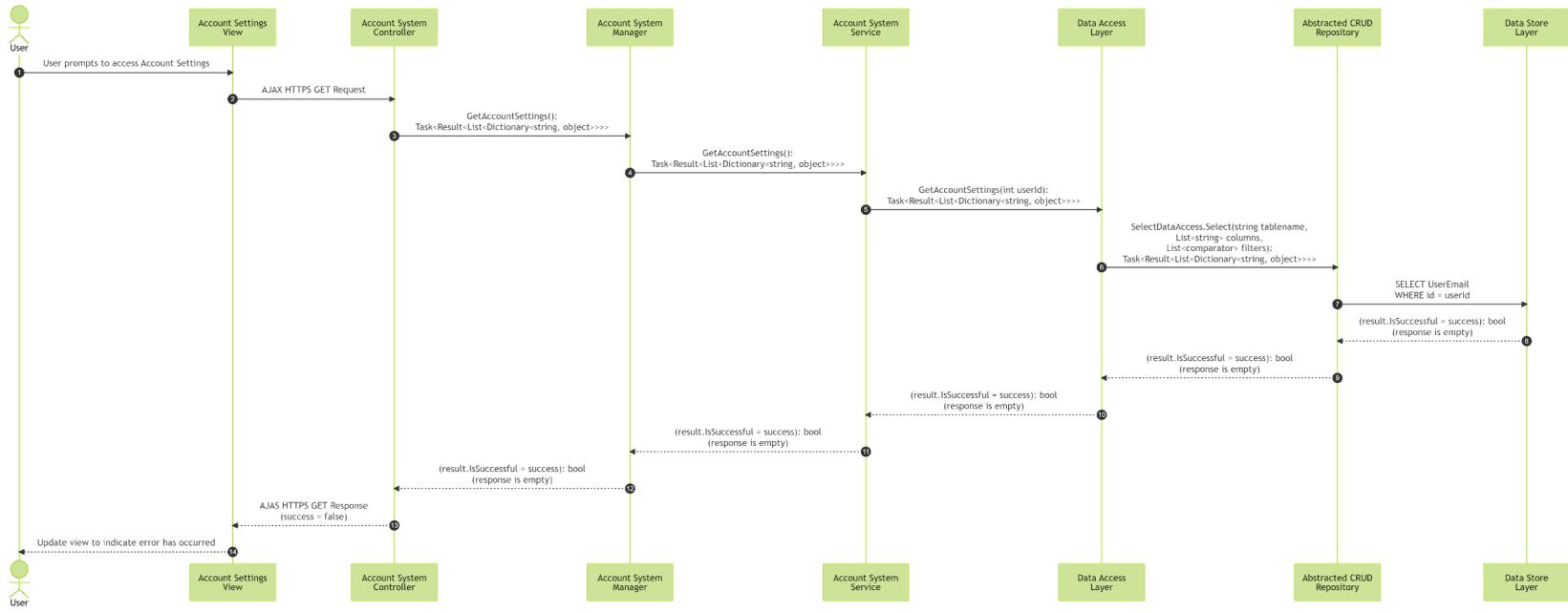
Failure Case: User is not able to load Account Settings View



Failure Case: Request is unsuccessful

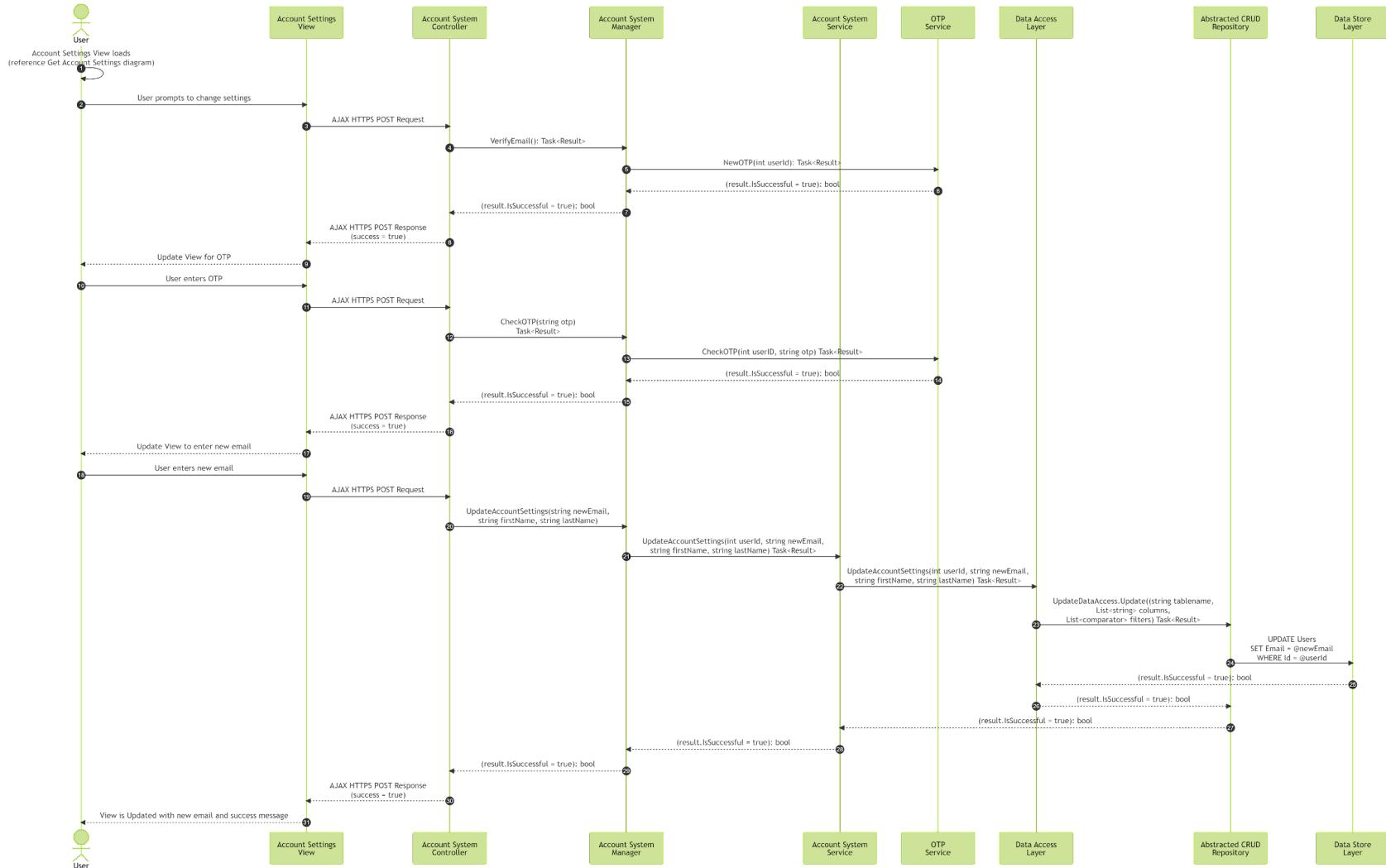


Failure Case: Response package is empty

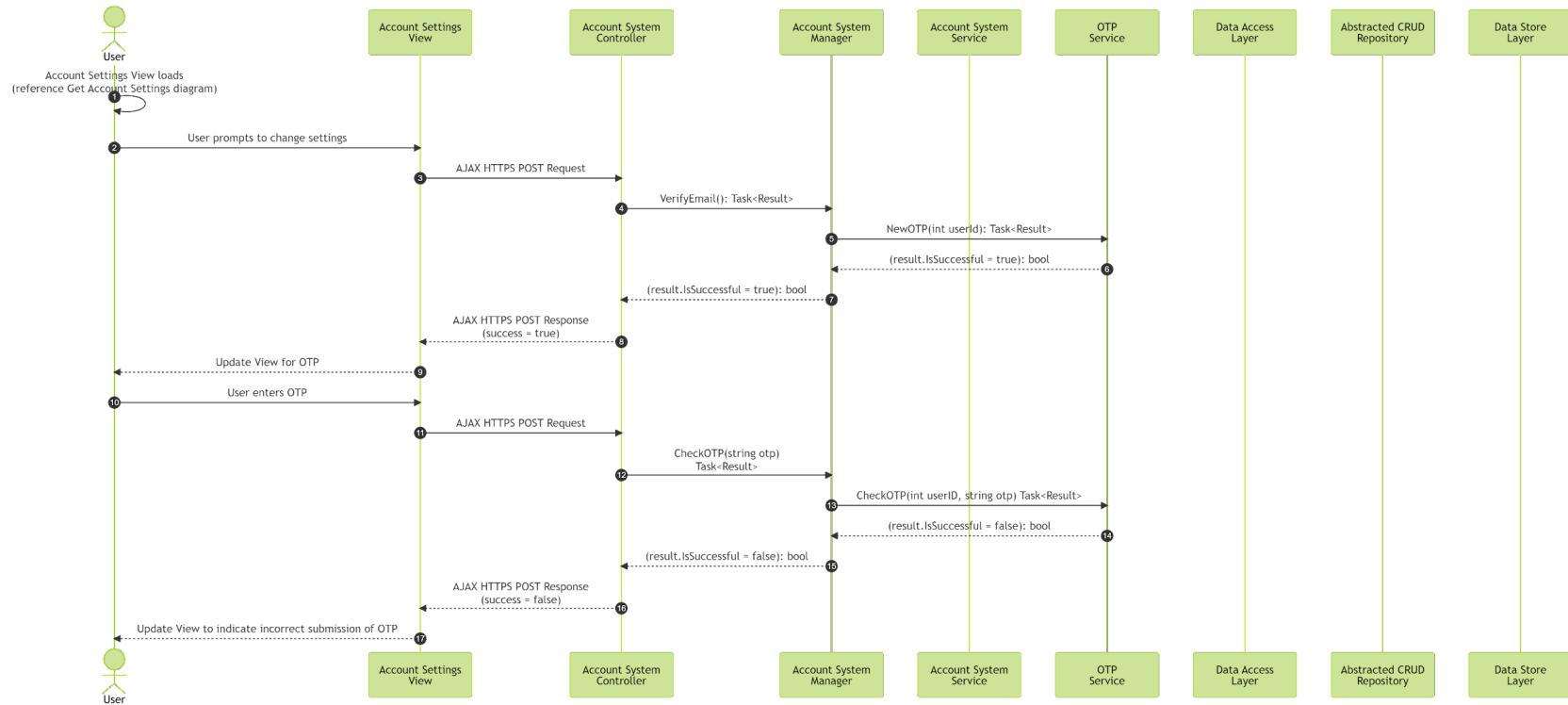


Change Account Settings (Email)

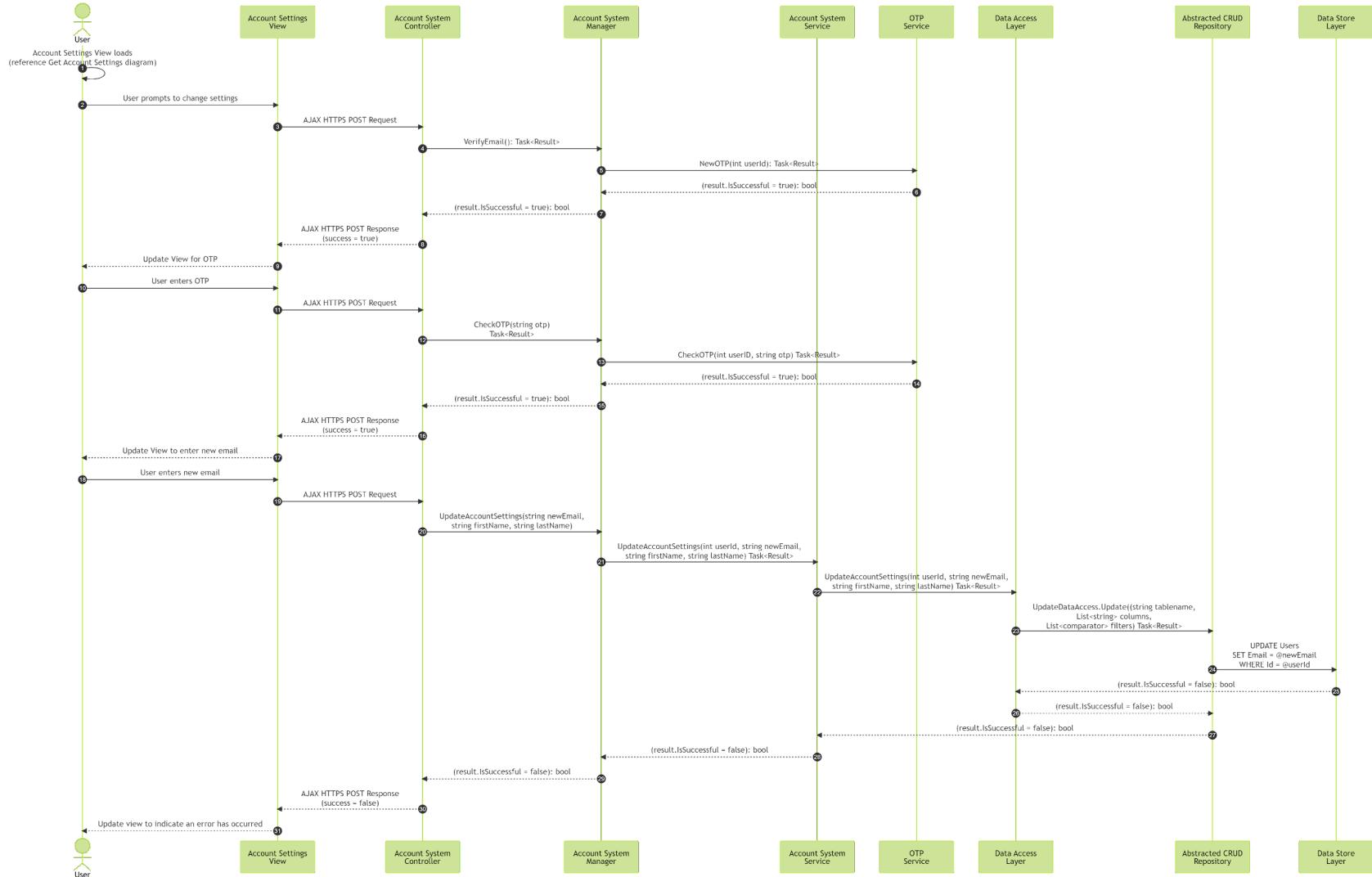
Success Case: User is able to change their email



Failure Case: User fails OTP

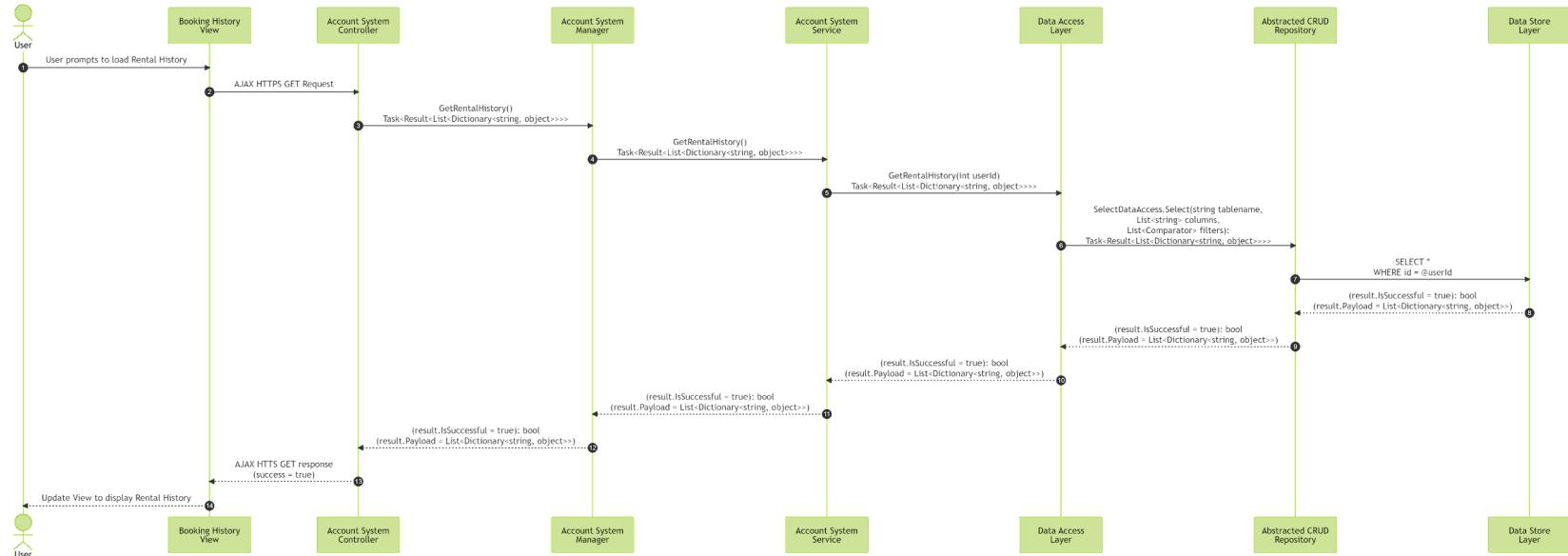


Failure Case: Changes are not saved in database

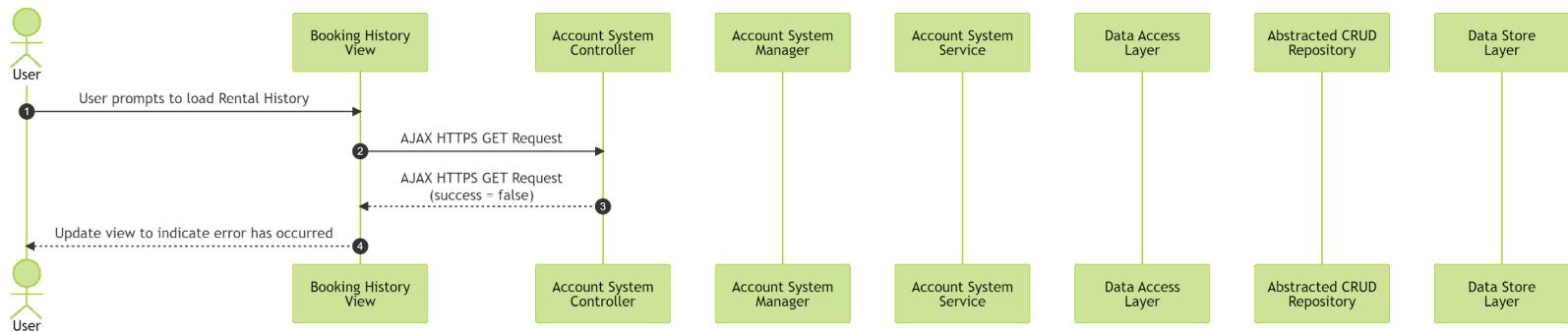


Get Rental History

Success Case: User is able to navigate to Booking History View

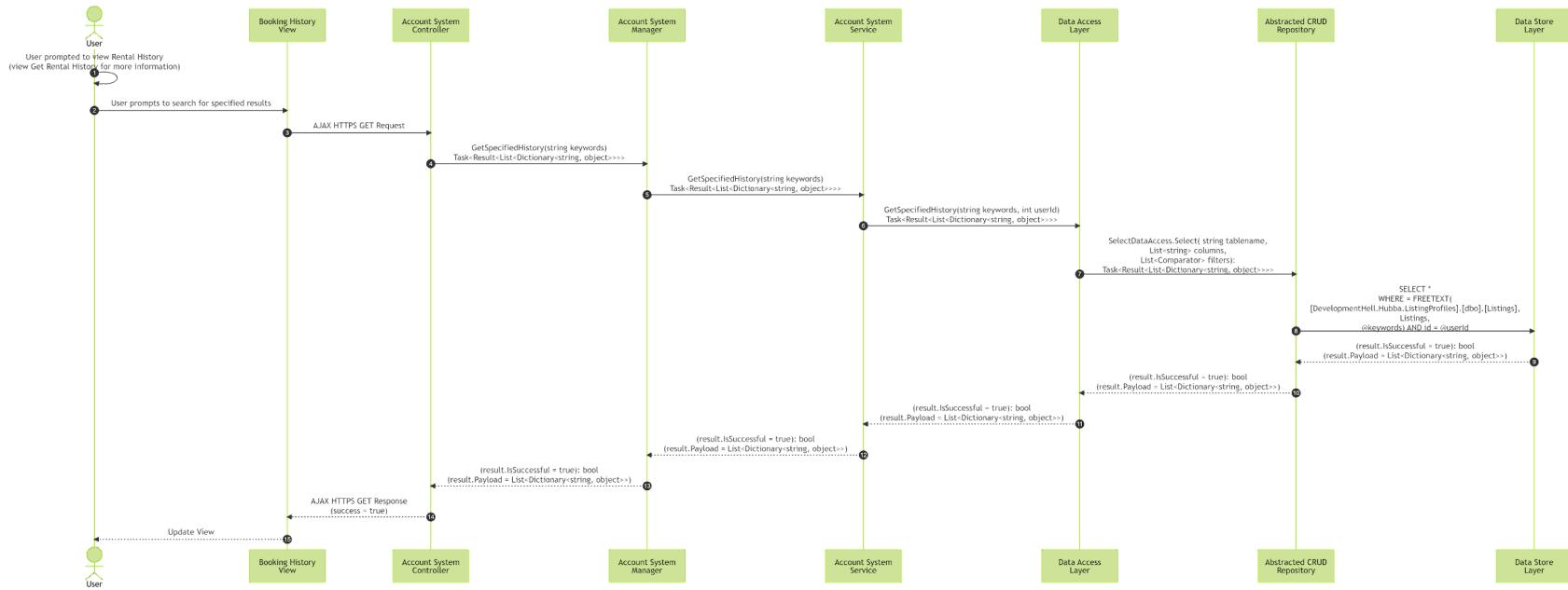


Failure Case: User is unable to access Rental History View

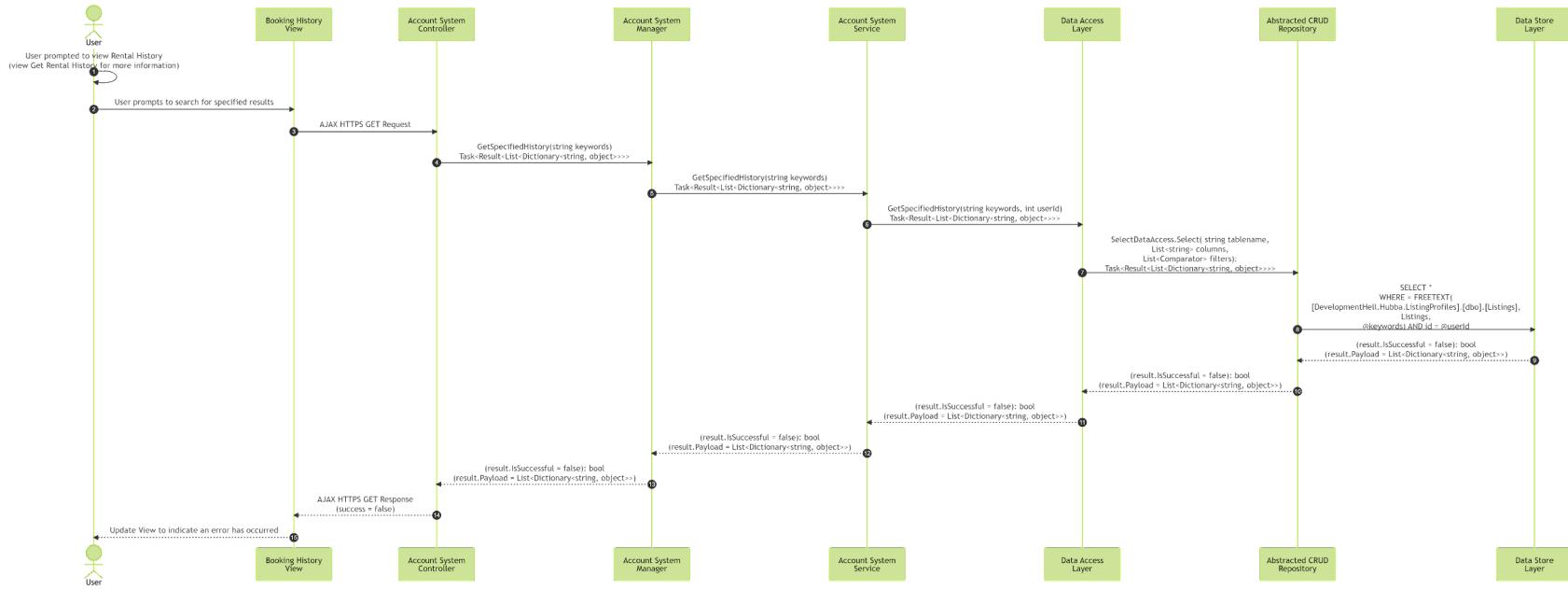


Search Rental History

Success Case: User is able to search for keywords to produce new results

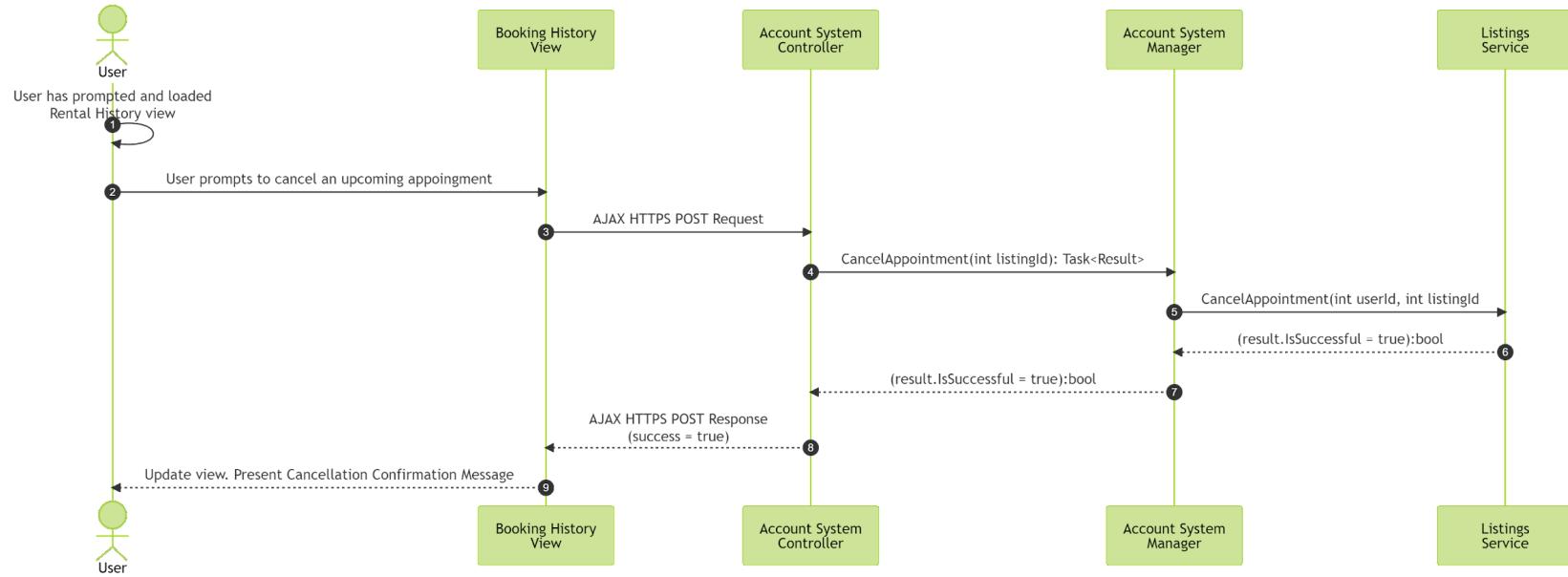


Failure Case: User's new results fail to load to view

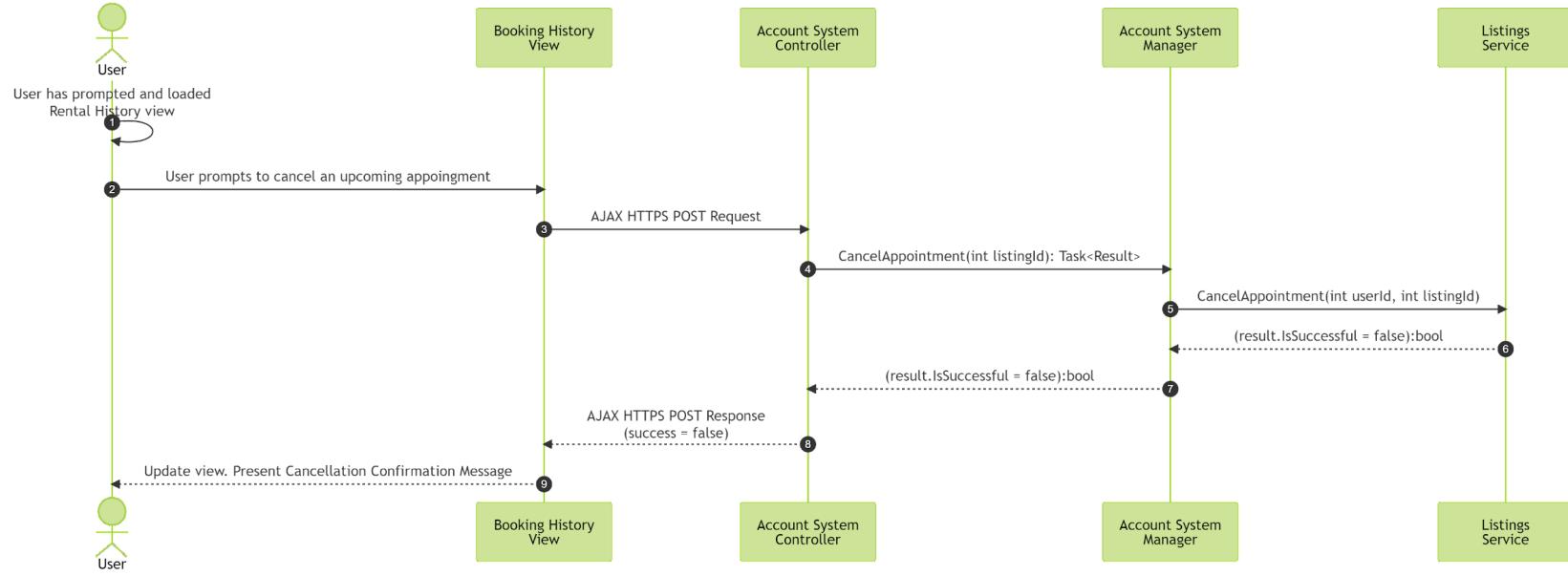


Booking Cancellation

Success Case: User is able to cancel a Booking Listing



Failure Case: Cancellation does not go through



Relational Table(s)

UserAccount Database

UserAccounts	Id	Email	PasswordHash	PasswordSalt	FirstName	LastName
	PK					

Listings Database

Listings	ListingID	OwnerID	UserID	---
	PK			

1..1

Bookings Database

Bookings	ListingID	BookingId	BookingStatusId
	FK	PK	
	1...*		