

---

# Software Design Specification

for

## Welcome Home

## Smart Escrow Contract

Version 1.0

Prepared by:

Casey Munga and Sergio Prieto

Smart Escrow Management

**April 26<sup>th</sup> 2022**

*Copyright © 2022 by Casey Munga and Sergio Prieto. All Rights reserved. This document cannot be shared distributed or its contents used in any way unless by express and written of the owners and developers of this product*

Copyright © 1994-1997 by Bradford D. Appleton. Permission is hereby granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies. (SDS sections)

Modified by Dr Renata Rand McFadden

# Table of Contents

<b>Table of Contents.....</b>	<b>ii</b>
<b>Revision History.....</b>	<b>iii</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope.....	1
1.5 References.....	1
<b>2. Overall Description.....</b>	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Functions.....	2
2.3 User Classes and Characteristics.....	2
2.4 Operating Environment.....	2
2.5 Design and Implementation Constraints.....	2
2.6 User Documentation.....	2
2.7 Assumptions and Dependencies.....	2
<b>3. External Interface Requirements.....</b>	<b>3</b>
3.1 User Interfaces Overview.....	3
3.2 Hardware Interfaces.....	3
3.3 Software Interfaces.....	3
<b>4. Design Considerations.....</b>	<b>3</b>
4.1 Assumptions and Dependencies.....	3
4.2 General Constraints.....	4
4.3 Development Methods.....	4
<b>5. System Architecture.....</b>	<b>4</b>
5.1 Architectural Strategies.....	4
5.2 High level Overview of System Architecture.....	5
<b>6. Human Interface Design.....</b>	<b>5</b>
6.1 Screen Images.....	5
6.2 Screen Objects and Actions.....	6
<b>7. Detailed System Design.....</b>	<b>6</b>
7.1 Data Structures.....	6
7.2 Component 1 Name.....	6
7.3 Component 2 Name.....	7
7.4 Component 3 Name.....	8
Appendix A: Glossary.....	8

## Revision History

Name	Date	Reason For Changes	Version
Casey Munga	04-26-2022	Initial Document	V1

# 1. Introduction

## 1.1 Purpose

*This is a Smart Contract for rental escrows using block chain technology. The smart contract solution will seek to reduce the need of trusted mediators, arbitration and enforcement costs when transacting rental property deposits, mitigating the risk of fraud and financial losses, as well as the reduction of malicious and accidental exceptions.*

## 1.2 Document Conventions

*None*

## 1.3 Intended Audience and Reading Suggestions

*This document's target audience is for the angel investor who are more technically including; the development team and stakeholders.*

## 1.4 Product Scope

### 1.4.1 Purpose

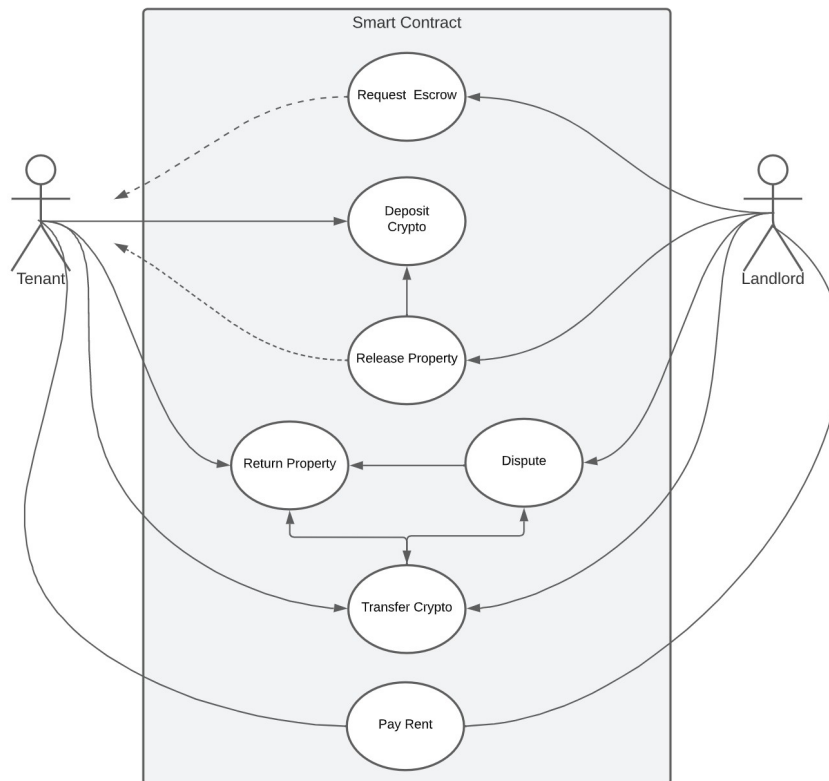
*When renting a property, the landlord and tenant are involved in a contract that requires a transaction in the form of a deposit. This money, in most cases, cannot be monitored by the tenant after it is given to the landlord.*

*In addition, the Deposit deductions are usually done arbitrarily by one the parties, without an efficient or cost effective means to dispute. By using an escrow smart contract, the risk of fraud can be reduced by ensuring proper escrow logic execution, along with the transparency of having the transaction accessible to all participants through an open distributed ledger or block-chain.*

### 1.4.2 High Level Flow

*This product is a new product that is designed to be scale-able web driven transactional app.*

Use Case Diagram : Sergio Prieto and Cassandra Munga



## 1.5 References

A **gateway** to the rental payment portal is listed below.

1. [Welcome Home Rental Payment Portal Smart Escrow Contract](#)

### Use Cases:

2. Project Overview
3. [Login](#)
4. [Initiate Contract](#)
5. [Create Contract](#)
6. [Create and Fill Wallet](#)
7. [Deploy Contract on Block Chain](#)

## 2. Overall Description

### 2.1 Product Perspective

*This product is a new product that is designed to be scale-able web driven transactional app .*

### 2.2 Product Functions

#### Major features

- Smart contract creation
- Crypto wallet creation
- Users views
- Accounts
- Dashboards
- Contracts
- Transaction
- Block Chain Deployment
- Arbitration module (Phase II)

### 2.3 User Classes and Characteristics

*The intended target market will be a class of real estate holders known as landlord and renters or tenants.*

## 2.4 Operating Environment

The Software will be hosted within the cloud with interactions with external entities namely financial institutions and Block-chain and MetaMask. Software environment will be a front end Web app that can accessed on a mobile device, tablet and laptop independent of any specific O/S.

## 2.5 Design and Implementation Constraints

At this time developers will be limited to the web and software as a service. As the needs demands a downloadable app for different OS may be developed. However that sits outside the present scope. MongoDB or AWS will be the DB system. HTTP and latest security protocols must be adhered to. Block-chain and cryptocurrency will the official trading currency. HTML5 for client , php for server response. Simple and unfettered interface. Site must be responsive.

## 2.6 User Documentation

Search and user help will be available inside the app. In app chat will be available inside the app during phase 2 but out of scope in this iteration. Use cases are listed in the document SmartContract-SRS-Munga\_Prieto.pdf

## 2.7 Assumptions and Dependencies

Selenium programming language, Truffle and block chain API s will be the main development and NoSQL., AWS Cloud

27 lines (27 sloc) | 1007 Bytes

```

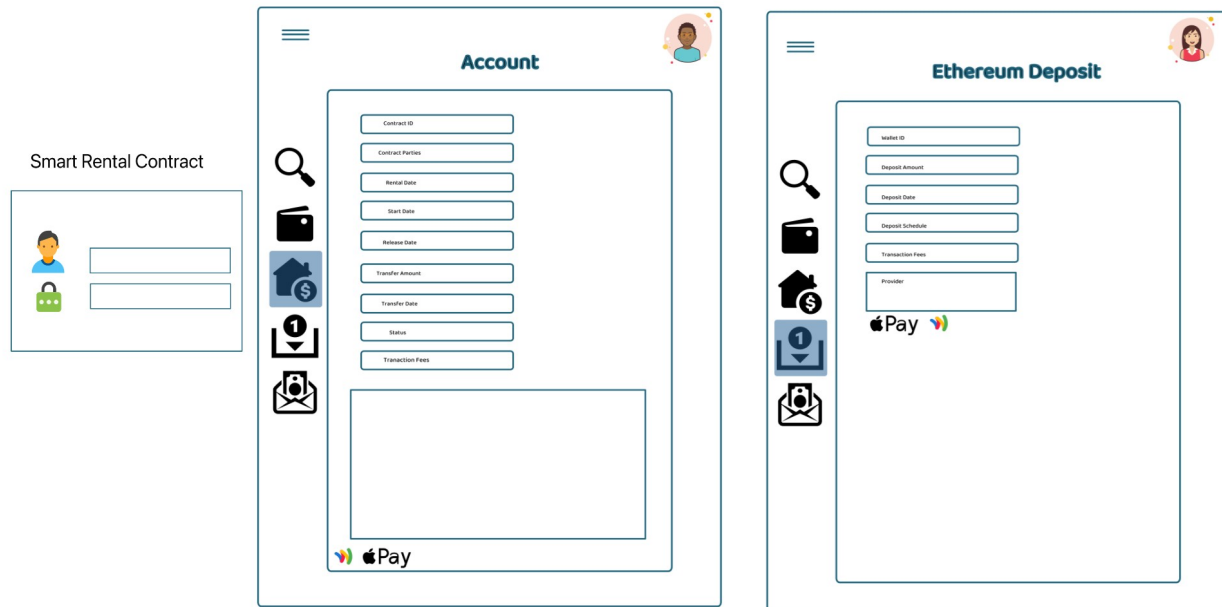
1  [
2    '{{repeat(5, 7)}}',
3    {
4      name: '{{firstName()}} {{surname()}}',
5      _clientId: '{{objectId()}}',
6      company: '{{company().toUpperCase()}}',
7      email: '{{email()}}',
8      password: '{{lorem(1, "words")}}',
9      index: '{{index()}}',
10     guid: '{{guid()}}',
11     isActive: '{{bool()}}',
12     contracts: [
13       '{{repeat(3)}}',
14       {
15         guid: '{{guid()}}',
16         rentalDate: '{{date(new Date(2014, 0, 1), new Date(), "YYYY-MM-ddThh:mm:ss Z")}}',
17         startDate: '{{date(new Date(2014, 0, 1), new Date(), "YYYY-MM-ddThh:mm:ss Z")}}',
18         releaseDate: '{{date(new Date(2014, 0, 1), new Date(), "YYYY-MM-ddThh:mm:ss Z")}}',
19         balance: '{{floating(1000, 4000, 2, "$0,0.00")}}',
20         transferDate: '{{date(new Date(2014, 0, 1), new Date(), "YYYY-MM-ddThh:mm:ss Z")}}',
21         transactionFees: '{{floating(10, 100, 2, "$0,0.00")}}',
22         days: '{{integer(1, 1000)}}',
23         status: '{{random("paid","deployed","refunded")}}'
24       }
25     ]
26   }
27 ]

```

### 3. External Interface Requirements

#### 3.1 User Interfaces Overview

GUI Model : Sergio Prieto and Cassandra Munga



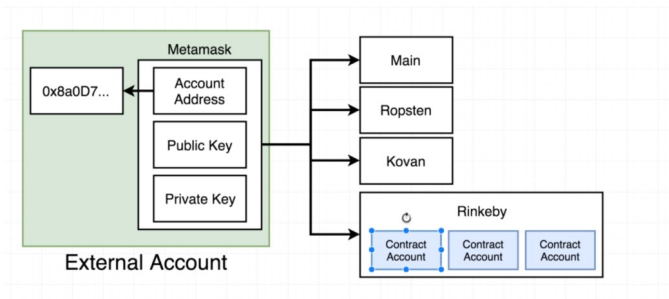
#### 3.2 Hardware Interfaces

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.*



### 3.3 Software Interfaces

Contract Account	
Field	Description
balance	Amount of ether this account owns
storage	Data storage for this contract
code	Raw machine code for this contract



## 4. Design Considerations

### 4.1 Assumptions and Dependencies

The following assumptions are

- smart and web capable device
- Will be cross platform tested
- End User do not have to have technical experience as the app will be simple
- Chat enabled in the future

## 4.2 General Constraints

*There are no major general constraints with the exception of a smart device such as a tablet, phone, or computer with the ability to receive network signals. The device must be able to access the web.*

## 4.3 Development Methods

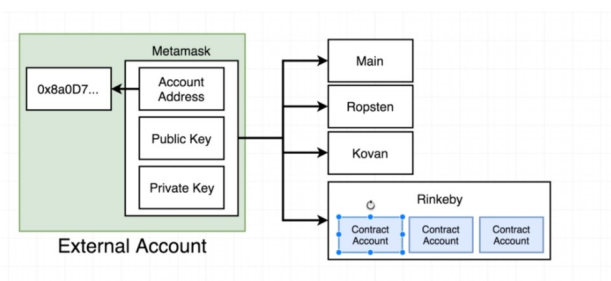
*The code development will be Selenium, Web 3, MetaMask used to develop the block chain technology HTML5, JavaScript, Node.js for the UI/UX, Java and object oriented coding will be used on the Server Side. The database will be NOSQL as MongoDB and Cassandra. Apps will be hosted and run in the cloud.*

## 5. System Architecture

### 5.1 Architectural Strategies

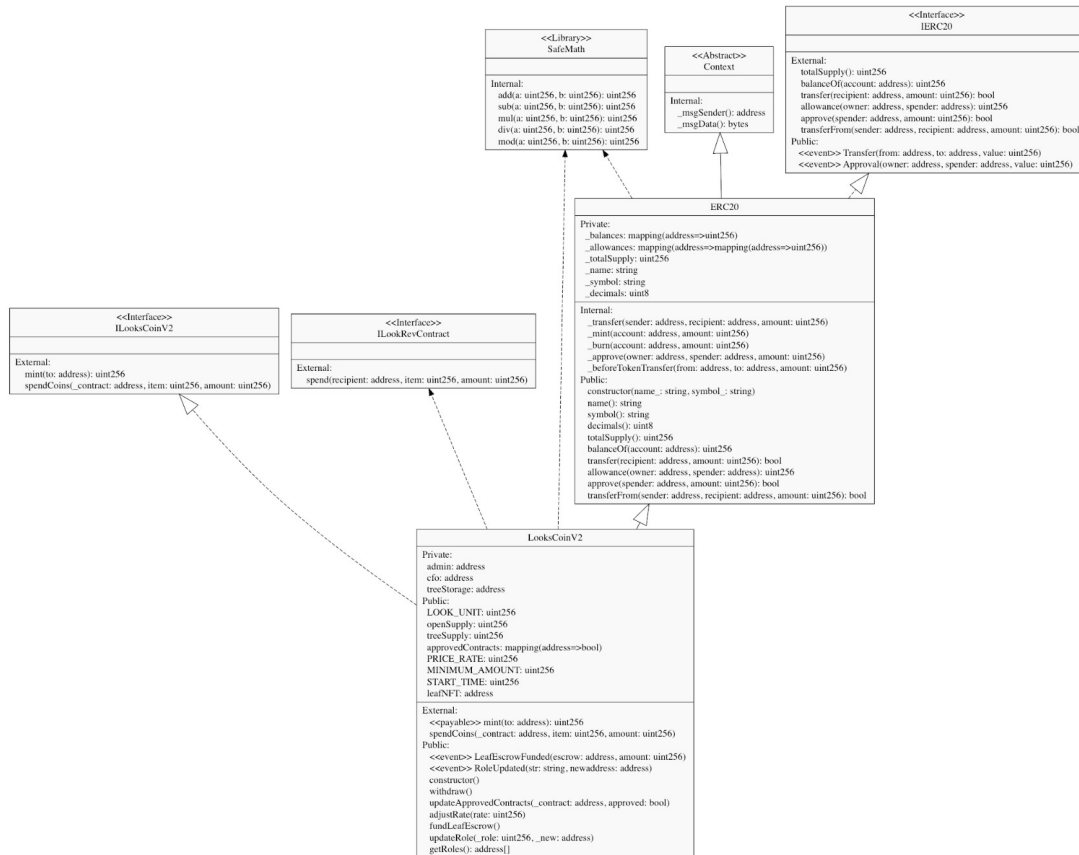
*Block Chain will be our primary architecture. Over AWS Server*

Contract Account	
Field	Description
balance	Amount of ether this account owns
storage	Data storage for this contract
code	Raw machine code for this contract



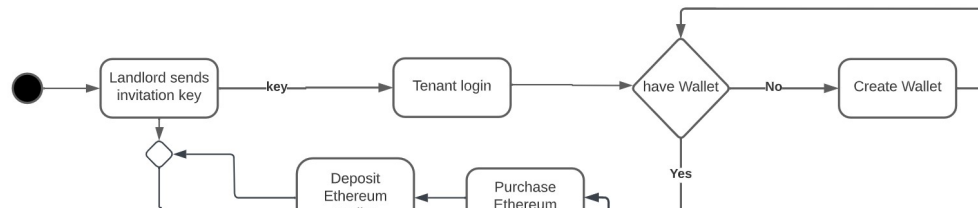
## 5.2 High level Overview of System Architecture

diagram showing the major subsystems and data repositories and their interconnections.

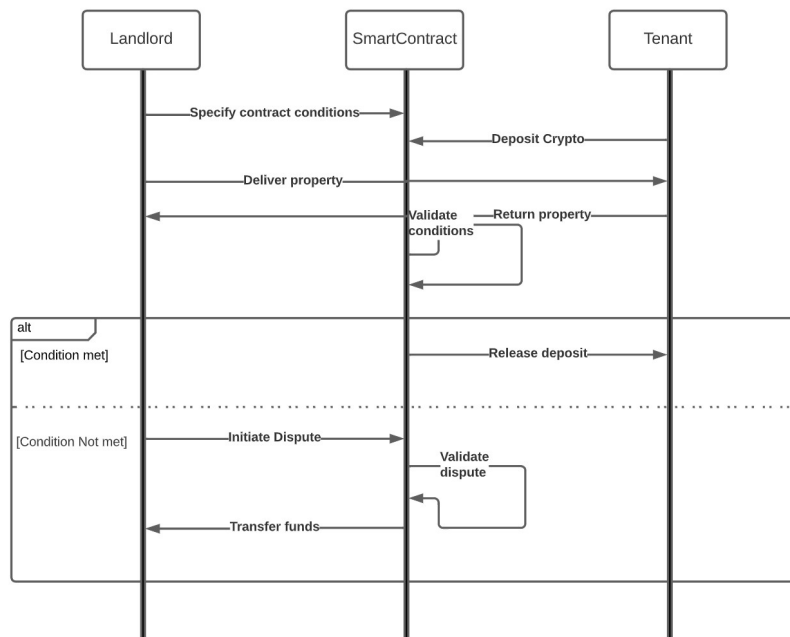


## SC0 -Smart Escrow - Activity Diagram

Casey Munga | April 19, 2022



Sequence Diagram : Sergio Prieto and Cassandra Munga

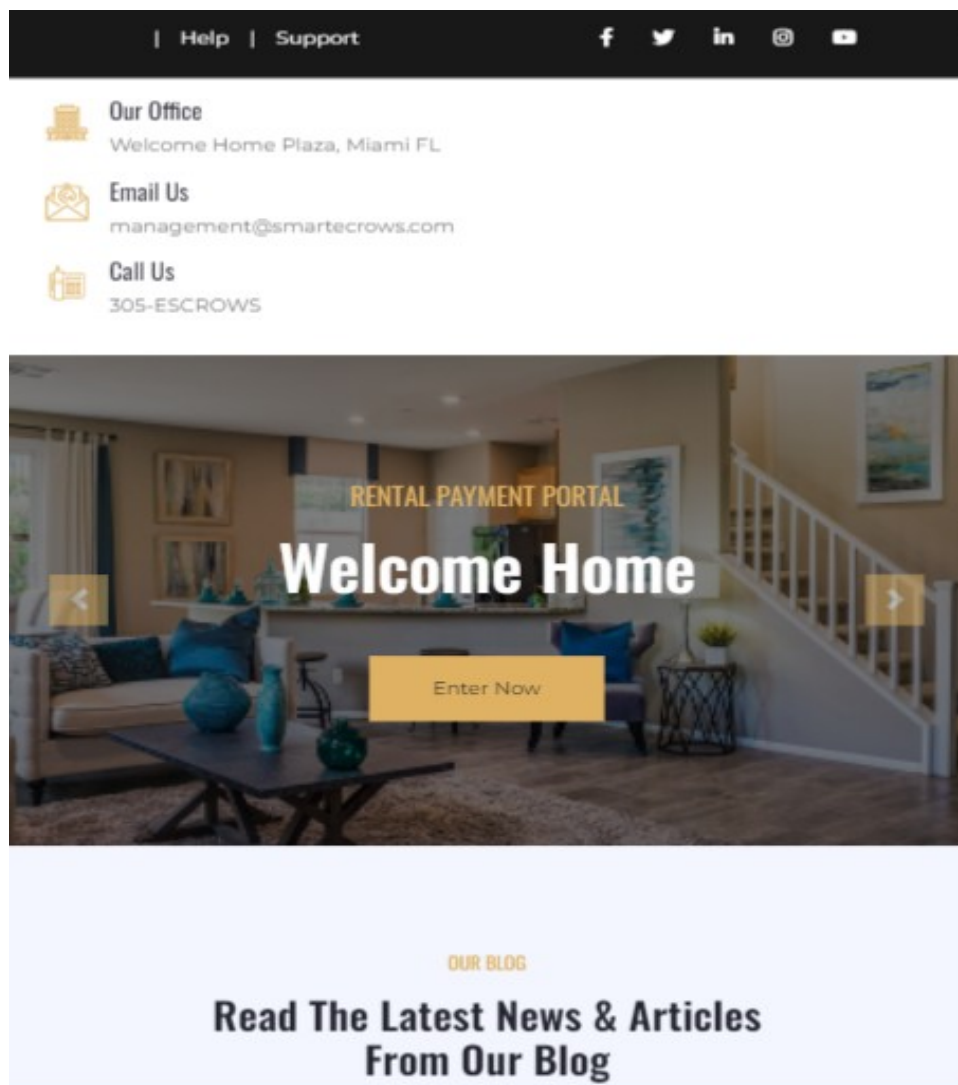


## 6. Human Interface Design

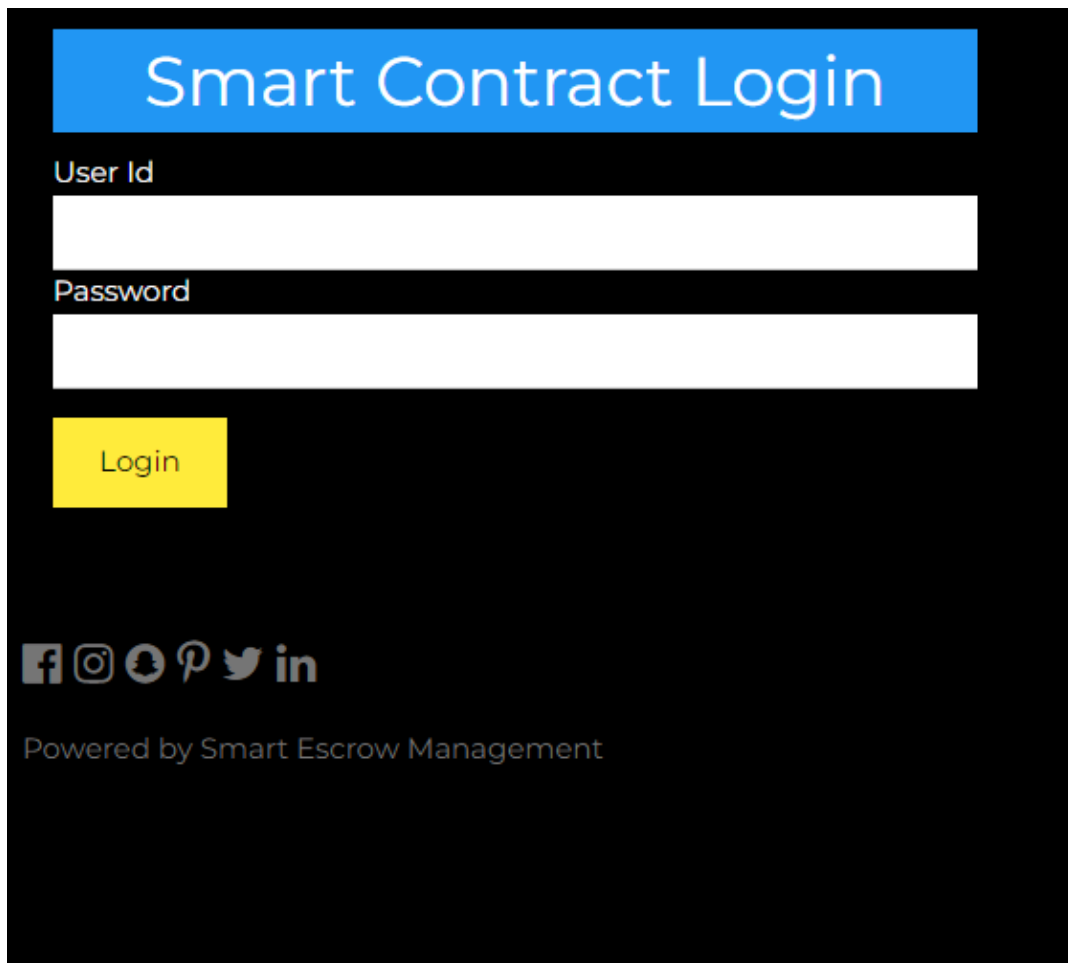
### 6.1 Screen Images

#### 6.1.0 Payment Portal Gateway

[Welcome Home Rental Portal Gateway](#)



### 6.1.1 Login



The image shows a login interface for a 'Smart Contract'. It features a blue header with the title 'Smart Contract Login'. Below this, there are two white input fields: one for 'User Id' and one for 'Password'. A yellow 'Login' button is positioned below the password field. At the bottom of the form, there is a row of social media icons (Facebook, Instagram, Snapchat, Pinterest, Twitter, and LinkedIn) and a footer text that reads 'Powered by Smart Escrow Management'.

Smart Contract Login

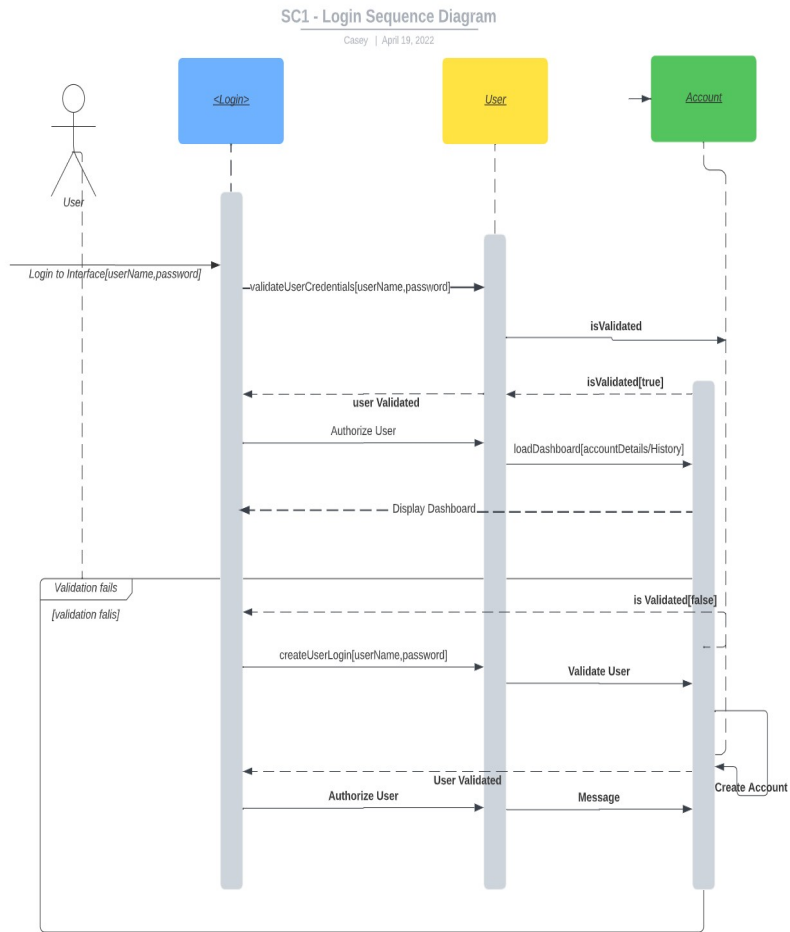
User Id

Password

Login

f i s p t in

Powered by Smart Escrow Management



[Login -full-size-image](#)



### 6.1.2 Initiate Contract

*This form page is only available to the Owner/Landlord. The landlord will enter the information and select the initiate Contract button. A notification email will be sent to the tenant.*

**New Smart Contract**

Contract  
s567b2346d237

Tenant Name  
Jane Doe

Tenant address  
janedoe@gmail.com

Rental Date  
mm/dd/yyyy

Start Date  
mm/dd/yyyy

Release Date  
mm/dd/yyyy

Transfer Amount  
\$1,000,000.00

Transfer Date  
mm/dd/yyyy

Status

Transaction Fees  
\$1,000,000.00

Contract Details

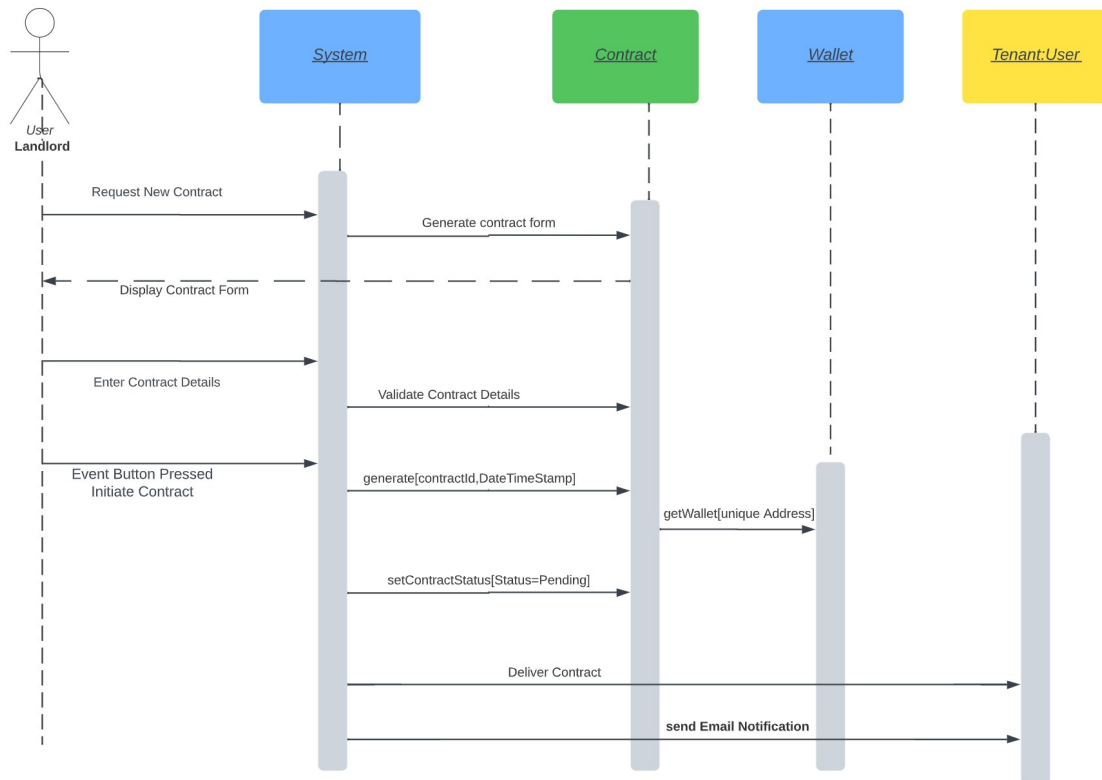
Initiate Contract

**Navigation Sidebar:**

- SEARCH
- HOME
- NEW
- HISTORY
- DEPOSIT
- LOGOUT

## SC2 - Create Contract Sequence Diagram

Casey Munga | April 19, 2022

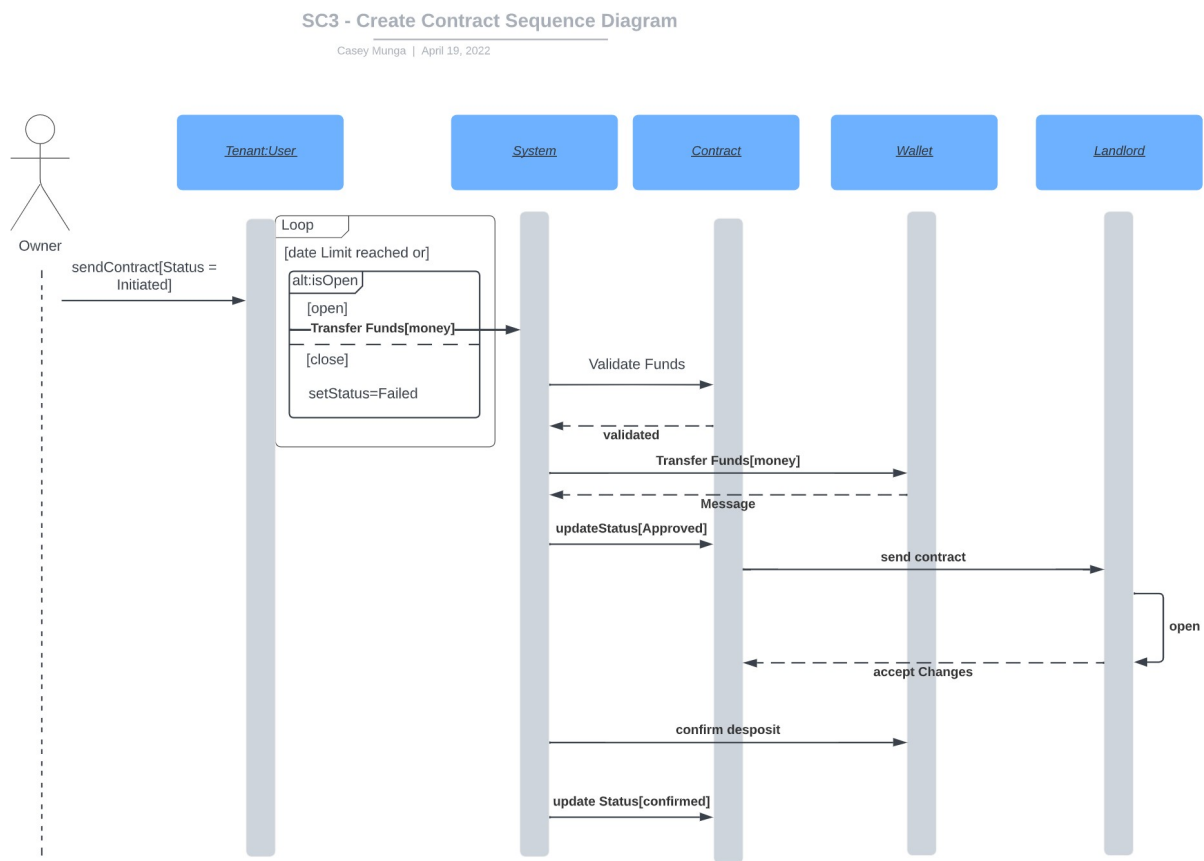
[Initiate Contract full-size-image](#)

### 6.1.3 Create Contracts

Owner receives an approved contract and will enter the transaction fees. The button changes to Confirm Contract.

The screenshot shows a mobile application interface for creating a new smart contract. On the left is a dark sidebar with navigation icons and labels: a hamburger menu, a magnifying glass labeled 'SEARCH', a house icon labeled 'HOME', a person icon labeled 'NEW', an eye icon labeled 'HISTORY', an envelope icon labeled 'DEPOSIT', and a right-pointing arrow labeled 'LOGOUT'. The main content area has a blue header 'New Smart Contract'. Below it are several input fields with labels: 'Contract' (containing 's567b2346d237'), 'Tenant Name' (containing 'Jane Doe'), 'Tenant address' (containing 'janedoe@gmail.com'), 'Rental Date' (placeholder 'mm/dd/yyyy' with a calendar icon), 'Start Date' (placeholder 'mm/dd/yyyy' with a calendar icon), 'Release Date' (placeholder 'mm/dd/yyyy' with a calendar icon), 'Transfer Amount' (containing '\$1,000,000.00'), 'Transfer Date' (placeholder 'mm/dd/yyyy' with a calendar icon), 'Status' (empty), 'Transaction Fees' (containing '\$1,000,000.00'), and 'Contract Details' (empty). At the bottom is a yellow button labeled 'Initiate Contract'.

New Smart Contract	
Contract	s567b2346d237
Tenant Name	Jane Doe
Tenant address	janedoe@gmail.com
Rental Date	mm/dd/yyyy
Start Date	mm/dd/yyyy
Release Date	mm/dd/yyyy
Transfer Amount	\$1,000,000.00
Transfer Date	mm/dd/yyyy
Status	
Transaction Fees	\$1,000,000.00
Contract Details	
Initiate Contract	



[Create Contract-full-size-image](#)

### 6.1.4 Account

View when tenant log on to account. Navigation Buttons will be present. Tenant is presented with view of any new Initiated contract or the current contract. See use case 1.5 Tenant selects Accept contract and is presented with deposit page to deposit money into wallet

The screenshot displays a user interface for a tenant's account. On the left is a dark sidebar with navigation icons and labels: a menu icon, a search icon labeled 'SEARCH', a home icon labeled 'HOME', an eye icon labeled 'HISTORY', an envelope icon labeled 'DEPOSIT', and a right arrow icon labeled 'LOGOUT'. The main content area has a blue header with the name 'Jane Doe'. Below this, the 'Contract' section shows a text field with the value 's567b2346d237'. The 'Rental Date' section shows '20/04/2022'. The 'Start Date' section shows '20/04/2022'. The 'Release Date' section shows '20/06/2023'. The 'Transfer Amount' section shows '0.00'. The 'Transfer Date' section is empty. The 'Status' section shows 'INITIATED'. The 'Transaction Fees' section shows '\$0.00'. The 'Contract Details' section is empty. At the bottom, there are two radio buttons: 'Accept Contract' (selected) and 'Decline Contract'. Below these is a yellow button labeled 'Update Contract'.

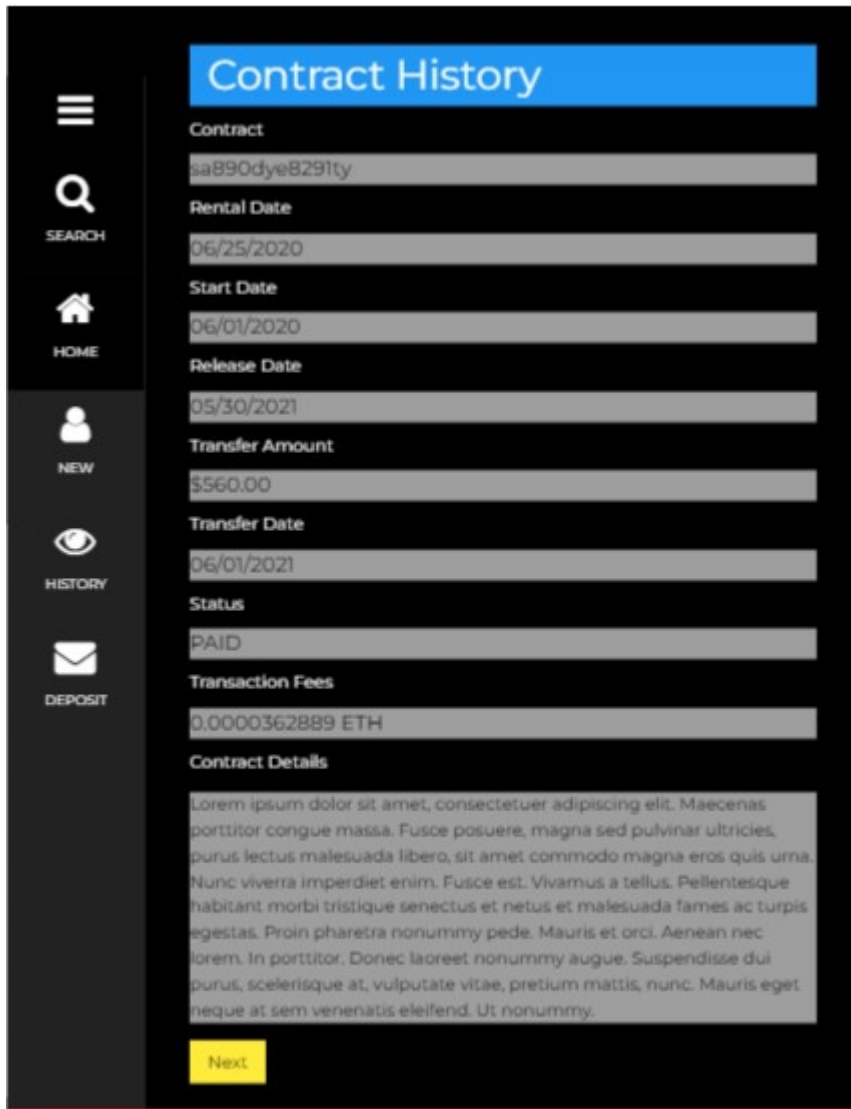
Field	Value
Contract	s567b2346d237
Rental Date	20/04/2022
Start Date	20/04/2022
Release Date	20/06/2023
Transfer Amount	0.00
Transfer Date	
Status	INITIATED
Transaction Fees	\$0.00
Contract Details	

☒ Accept Contract ☐ Decline Contract

Update Contract

### 6.1.5 History

User is shown most recent contract transaction but can opt to see a list of all historical data of every transaction

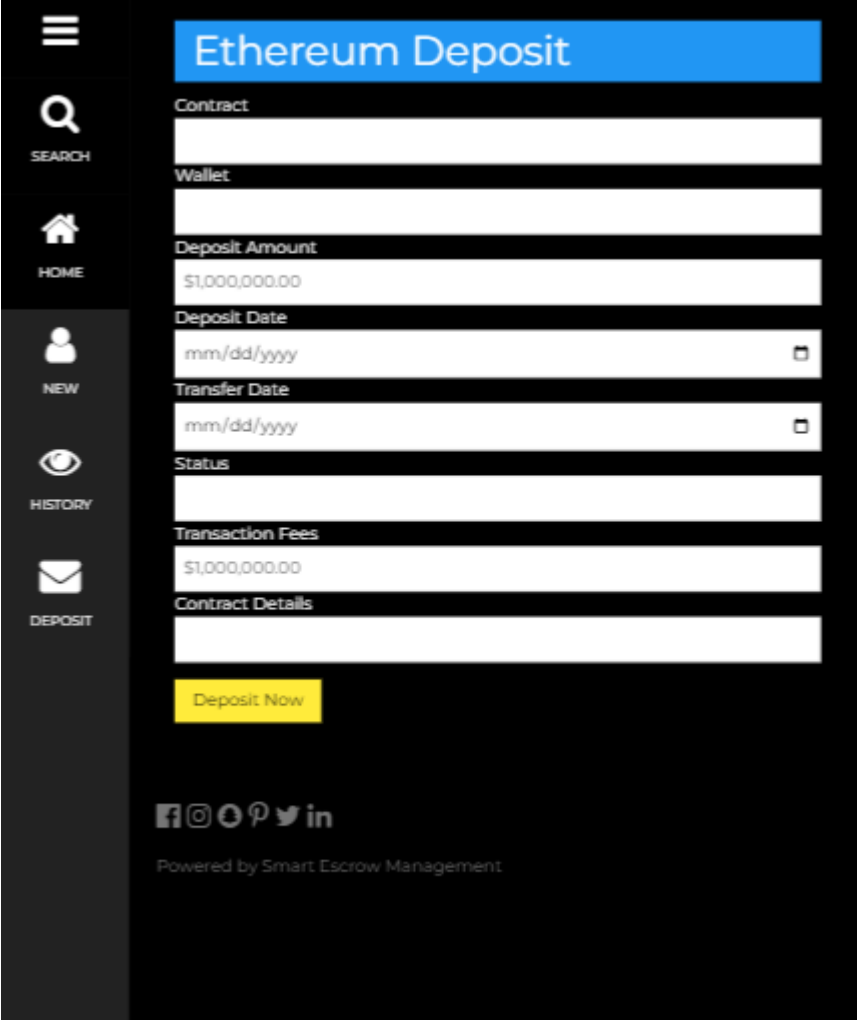


The image shows a user interface for a 'Contract History' page. On the left is a dark sidebar with navigation icons and labels: a hamburger menu, a magnifying glass labeled 'SEARCH', a house icon labeled 'HOME', a person icon labeled 'NEW', an eye icon labeled 'HISTORY', and an envelope icon labeled 'DEPOSIT'. The main content area has a blue header 'Contract History'. Below it, a table displays contract details. The table has a 'Contract' row with the value 'sa890dye8291ty', and rows for 'Rental Date' (06/25/2020), 'Start Date' (06/01/2020), 'Release Date' (05/30/2021), 'Transfer Amount' (\$560.00), 'Transfer Date' (06/01/2021), 'Status' (PAID), and 'Transaction Fees' (0.0000362889 ETH). Below the table is a 'Contract Details' section containing a block of Lorem Ipsum text. At the bottom right of the details section is a yellow 'Next' button.

Contract History	
Contract	sa890dye8291ty
Rental Date	06/25/2020
Start Date	06/01/2020
Release Date	05/30/2021
Transfer Amount	\$560.00
Transfer Date	06/01/2021
Status	PAID
Transaction Fees	0.0000362889 ETH
<strong>Contract Details</strong>	
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci. Aenean nec lorem. In porttitor. Donec laoreet nonummy augue. Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.	
<a href="#">Next</a>	

### 6.1.6 Deposit

*This page is served when the user wants to deposit money from a banking institution into his wallet and purchase ethereum.*



The image shows a mobile application interface for an "Ethereum Deposit" form. The interface has a dark theme with a sidebar on the left containing navigation icons: a menu icon, a search icon labeled "SEARCH", a home icon labeled "HOME", a user profile icon labeled "NEW", an eye icon labeled "HISTORY", and an envelope icon labeled "DEPOSIT". The main content area has a blue header with the title "Ethereum Deposit". Below the header, there are several input fields: "Contract" (empty), "Wallet" (empty), "Deposit Amount" (pre-filled with "\$1,000,000.00"), "Deposit Date" (placeholder "mm/dd/yyyy" with a calendar icon), "Transfer Date" (placeholder "mm/dd/yyyy" with a calendar icon), "Status" (empty), "Transaction Fees" (pre-filled with "\$1,000,000.00"), and "Contract Details" (empty). A yellow "Deposit Now" button is located below the input fields. At the bottom of the form, there are social media icons for Facebook, Instagram, YouTube, Pinterest, and LinkedIn, followed by the text "Powered by Smart Escrow Management".

**Ethereum Deposit**

Contract

Wallet

Deposit Amount

\$1,000,000.00

Deposit Date

mm/dd/yyyy

Transfer Date

mm/dd/yyyy

Status

Transaction Fees

\$1,000,000.00

Contract Details

Deposit Now

Facebook Instagram YouTube Pinterest LinkedIn

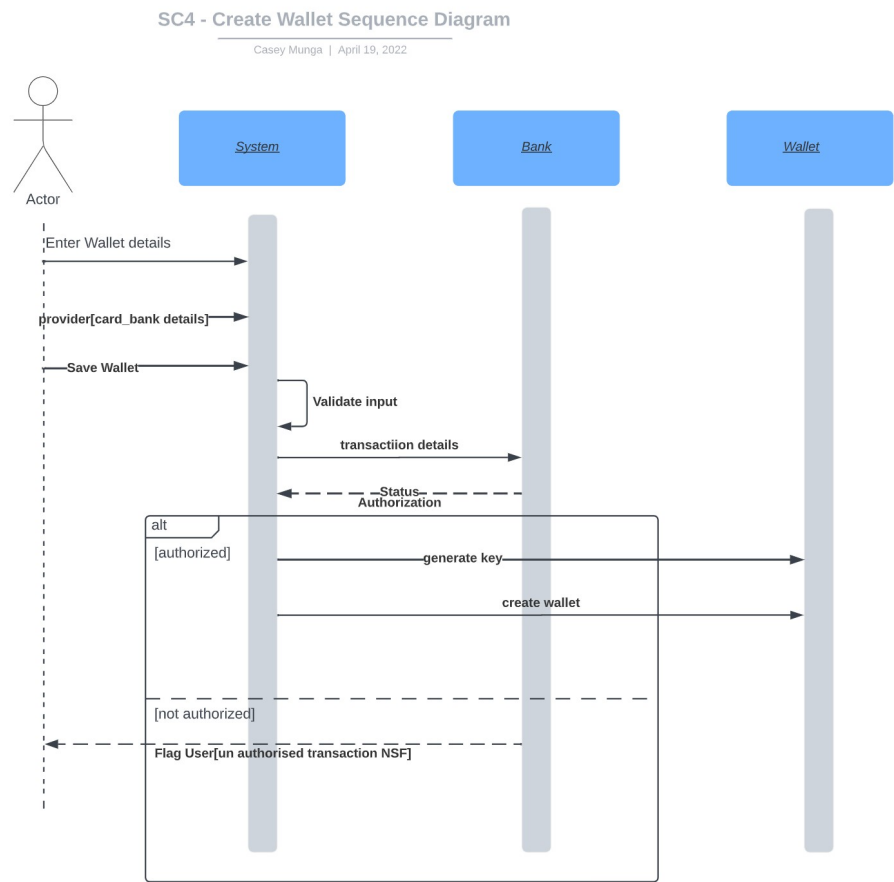
Powered by Smart Escrow Management

### 6.1.7 Create Wallet

All currency must be kept in a wallet. New users must create a wallet and assign an external entity such as a credit card, bank account, paypal and must transfer / deposit money into their account at this stage. The wallet is a transactional component and will be used to uniquely identify the account holder. An account can only have one wallet, but a wallet can be used in more than one account. This form will be repurposed programmatically to create wallet when served up

The screenshot shows a web application interface for an 'Ethereum Deposit'. On the left is a dark sidebar with navigation icons and labels: a menu icon, a search icon labeled 'SEARCH', a home icon labeled 'HOME', a person icon labeled 'NEW', an eye icon labeled 'HISTORY', and an envelope icon labeled 'DEPOSIT'. The main content area has a blue header 'Ethereum Deposit'. Below the header are several form fields: 'Contract' (empty), 'Wallet' (empty), 'Deposit Amount' (pre-filled with '\$1,000,000.00'), 'Deposit Date' (placeholder 'mm/dd/yyyy' with a calendar icon), 'Transfer Date' (placeholder 'mm/dd/yyyy' with a calendar icon), 'Status' (empty), 'Transaction Fees' (pre-filled with '\$1,000,000.00'), and 'Contract Details' (empty). A yellow 'Deposit Now' button is located below the form fields. At the bottom of the main area are social media icons for Facebook, Instagram, YouTube, Pinterest, Twitter, and LinkedIn, followed by the text 'Powered by Smart Escrow Management'.

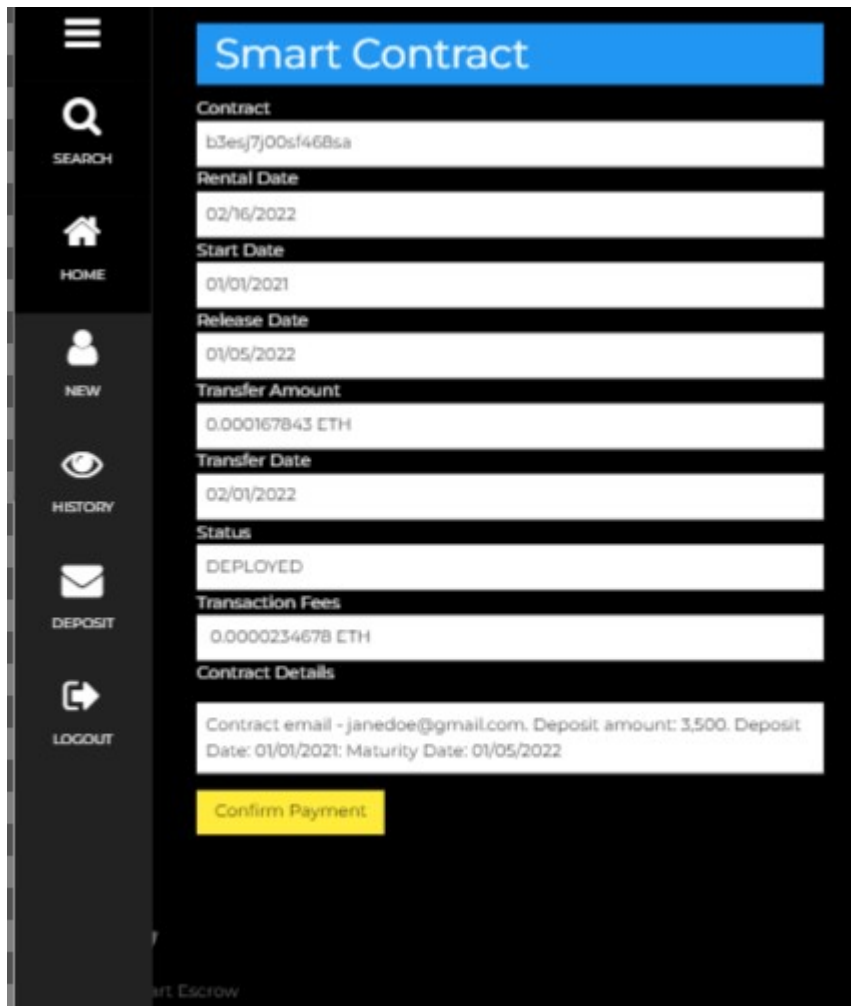




[Create Wallet full-size-image](#)

### 6.1.8 Deploy Contract

This page is available after the contract had been confirmed and all fees paid. The button's text property will be changed to Deployed. The owner can now deploy the account to the block chain. Please see use case section 1.5



The image shows a web interface for a 'Smart Contract'. On the left is a dark sidebar with navigation icons and labels: a menu icon, a search icon labeled 'SEARCH', a home icon labeled 'HOME', a user icon labeled 'NEW', an eye icon labeled 'HISTORY', an envelope icon labeled 'DEPOSIT', and a logout icon labeled 'LOGOUT'. The main content area has a blue header 'Smart Contract'. Below it, several fields are displayed with labels and values: 'Contract' (b3esj7j00sf46Bsa), 'Rental Date' (02/16/2022), 'Start Date' (01/01/2021), 'Release Date' (01/05/2022), 'Transfer Amount' (0.000167843 ETH), 'Transfer Date' (02/01/2022), 'Status' (DEPLOYED), and 'Transaction Fees' (0.0000234678 ETH). A section titled 'Contract Details' contains the text: 'Contract email - janedoe@gmail.com. Deposit amount: 3,500. Deposit Date: 01/01/2021: Maturity Date: 01/05/2022'. At the bottom of this section is a yellow button labeled 'Confirm Payment'.

Field	Value
Contract	b3esj7j00sf46Bsa
Rental Date	02/16/2022
Start Date	01/01/2021
Release Date	01/05/2022
Transfer Amount	0.000167843 ETH
Transfer Date	02/01/2022
Status	DEPLOYED
Transaction Fees	0.0000234678 ETH

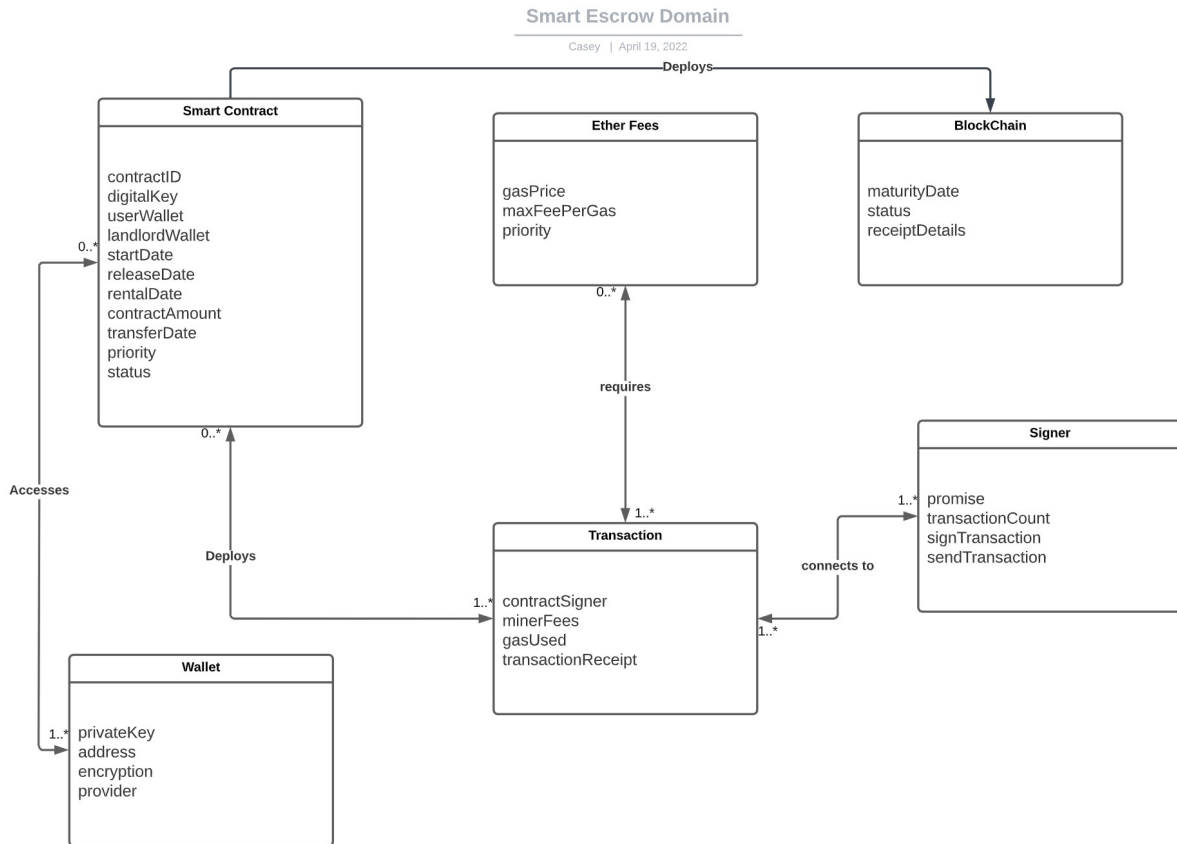
**Contract Details**

Contract email - janedoe@gmail.com. Deposit amount: 3,500. Deposit Date: 01/01/2021: Maturity Date: 01/05/2022

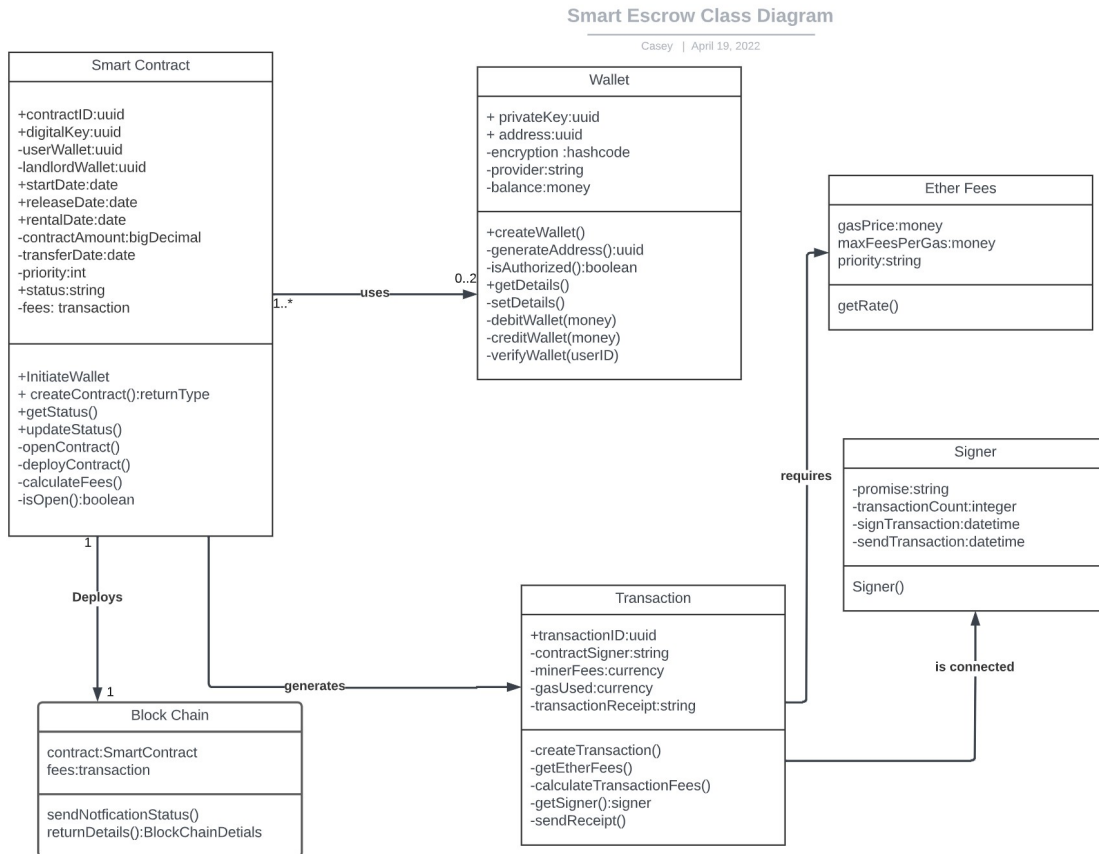
**Confirm Payment**

## 7. Detailed System Design

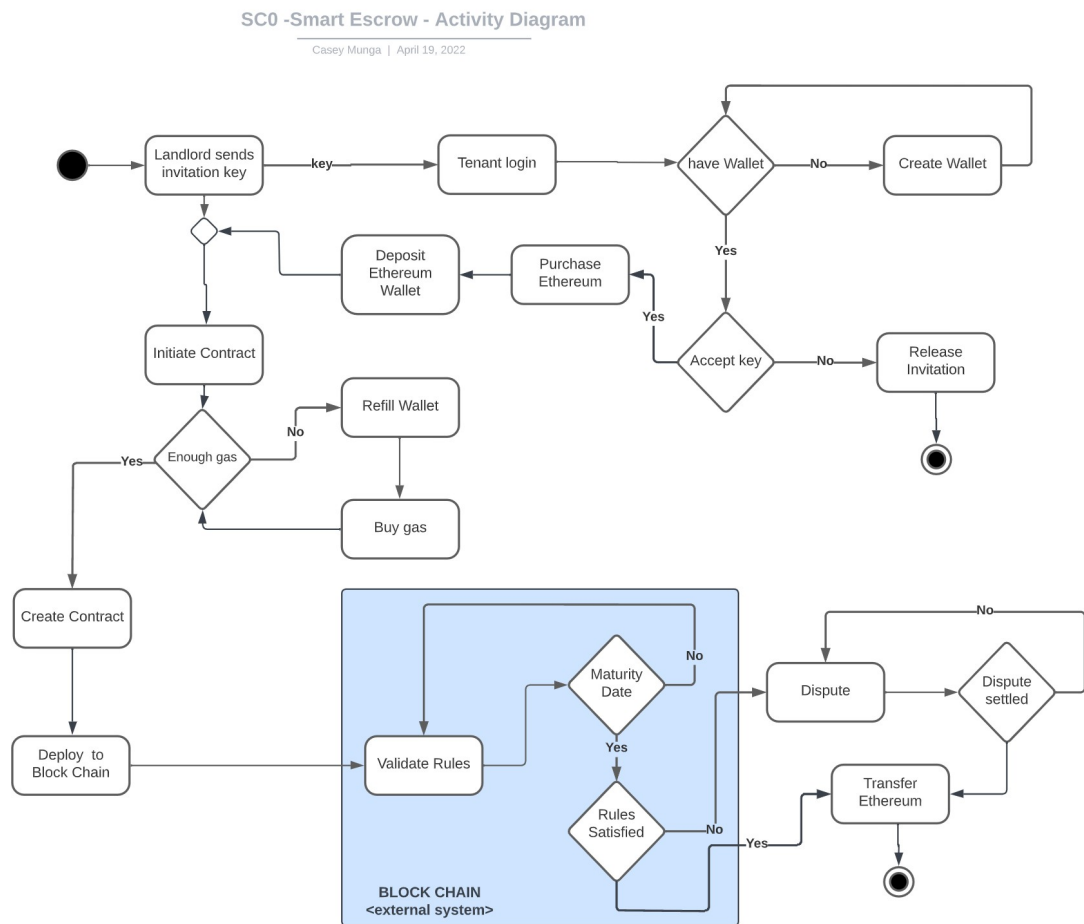
### 7.1.1 Smart Escrow Domain Diagram



### 7.1.2 Smart Escrow Class Diagram

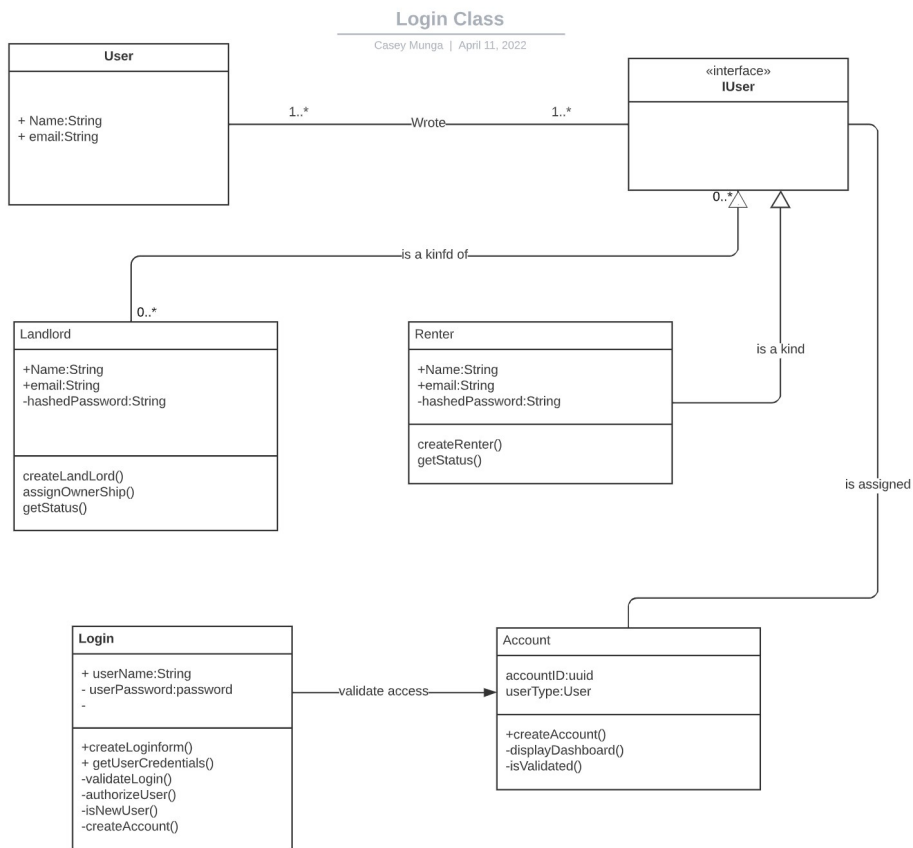


### 7.1.3 High Level View Of System

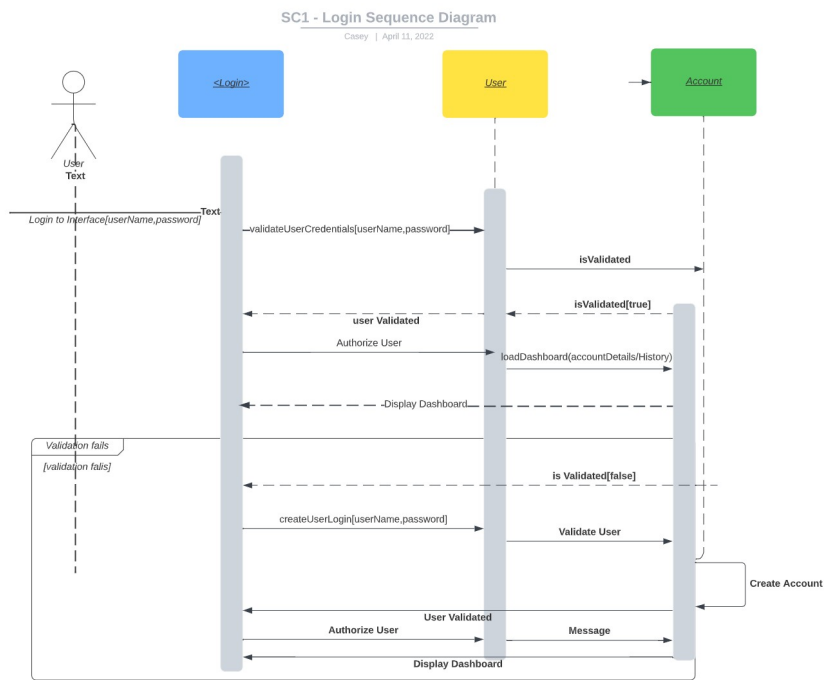


## 7.2 Login

### 7.2.1 Class Diagram

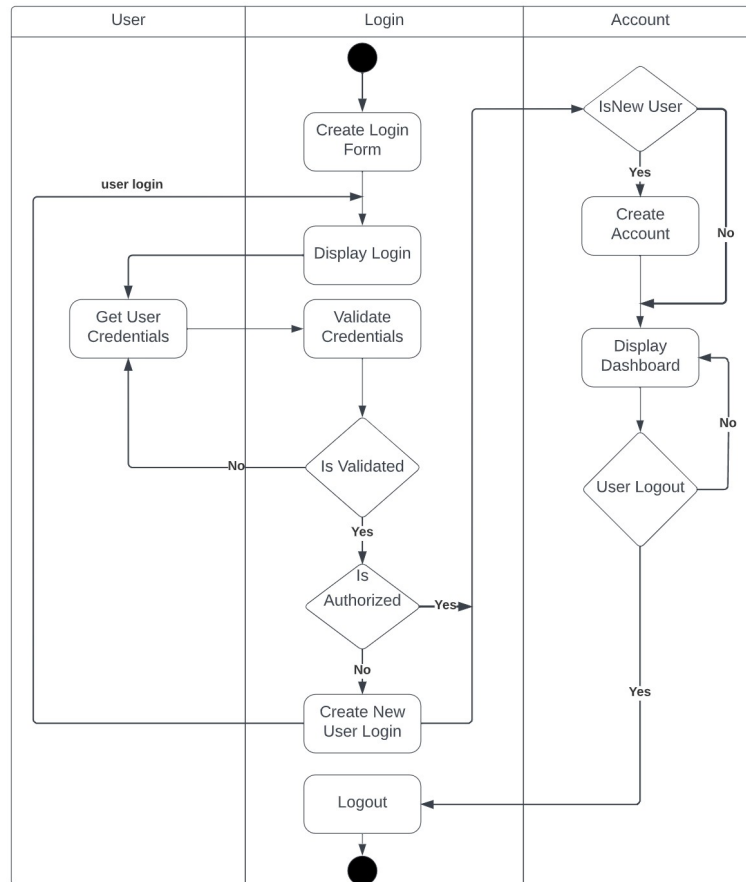


## 7.2.2 Activity Diagram



## SC1 - Login Activity Diagram

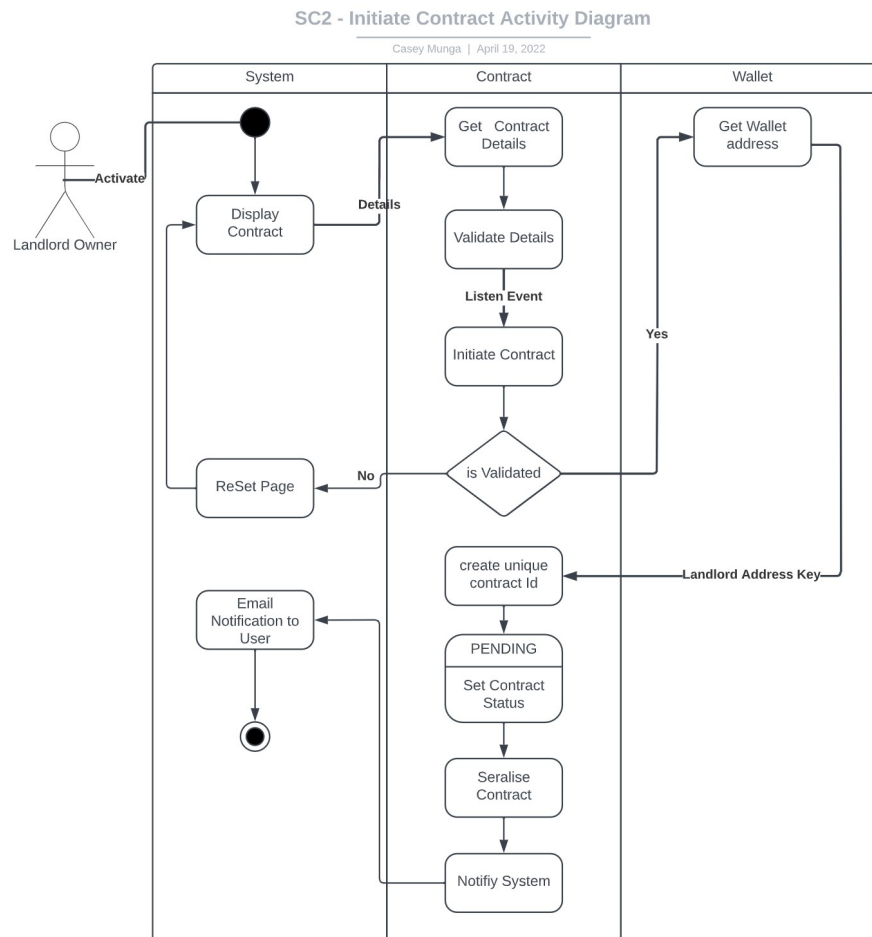
Casey Munga | April 19, 2022



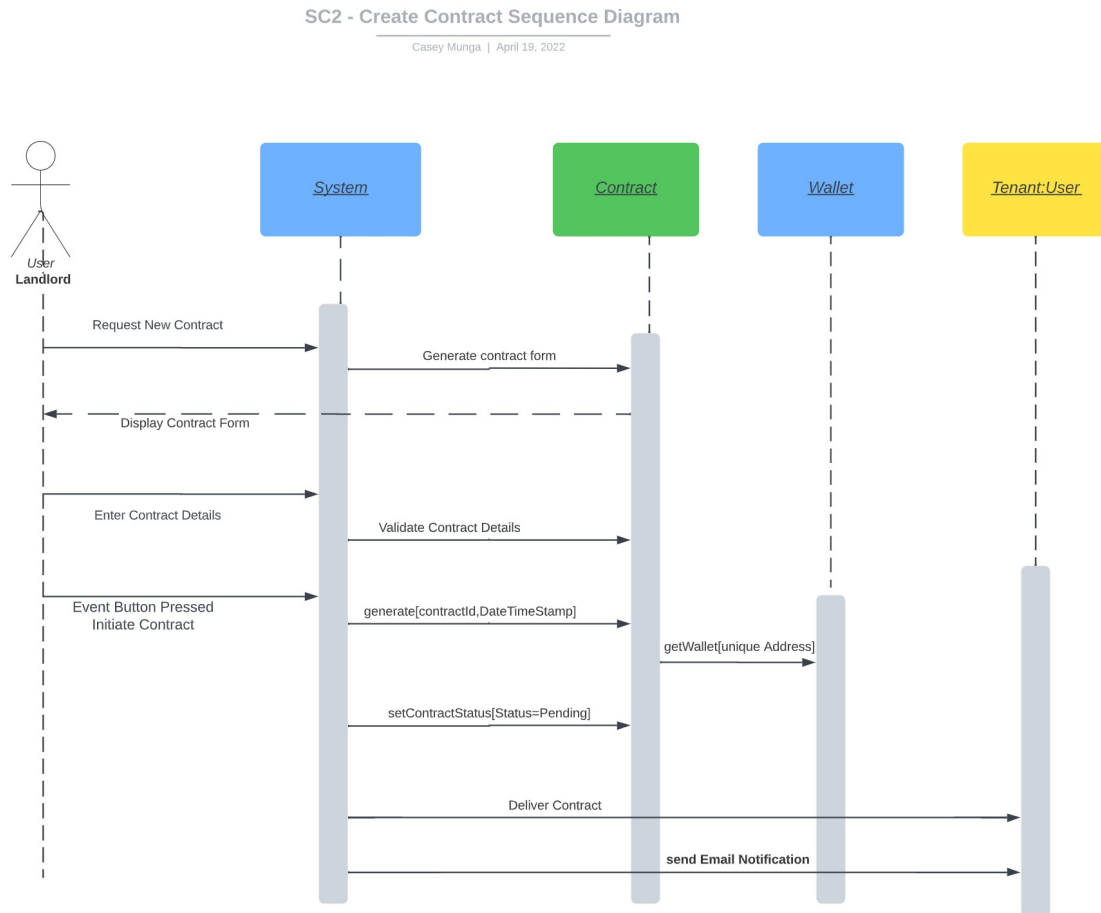


## 7.3 Initiate Contract

### 7.3.1 Activity Diagram

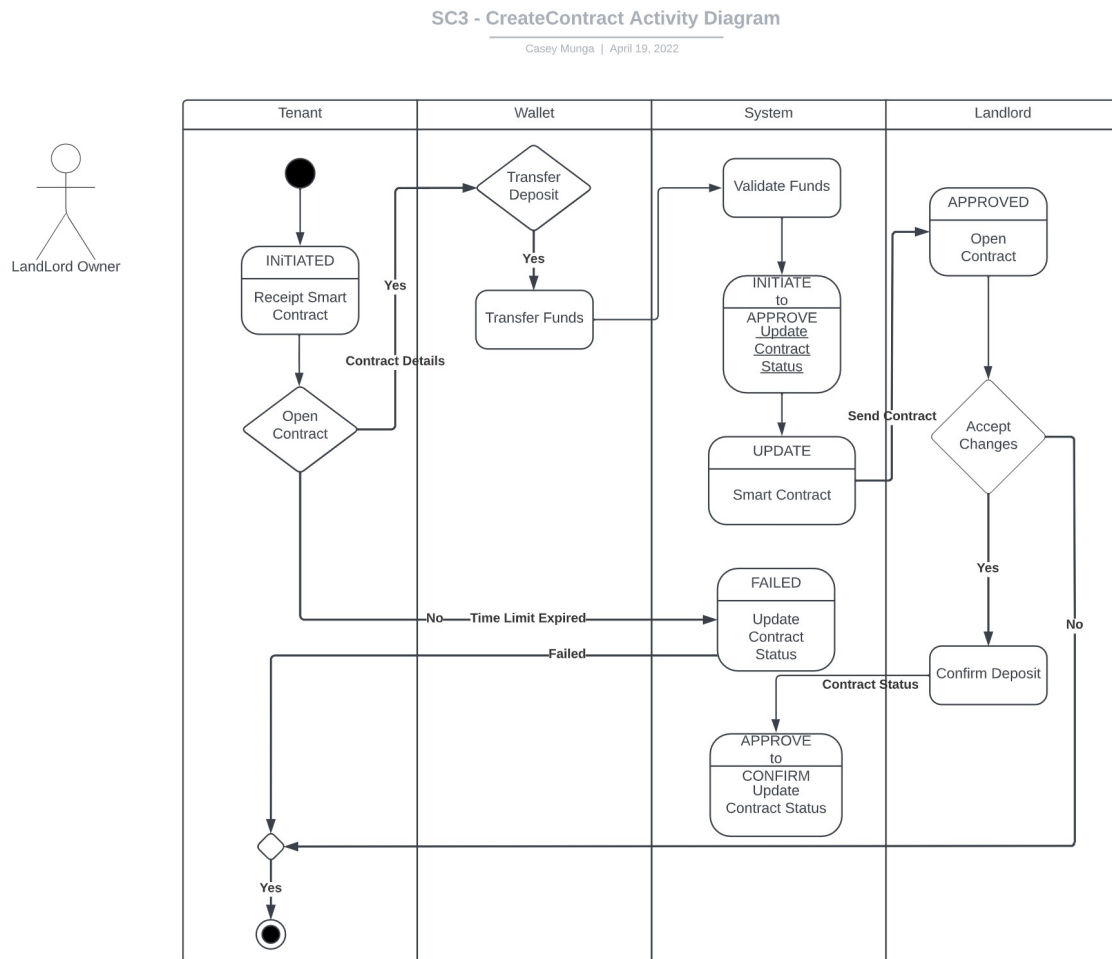


### 7.3.2 Sequence Diagram

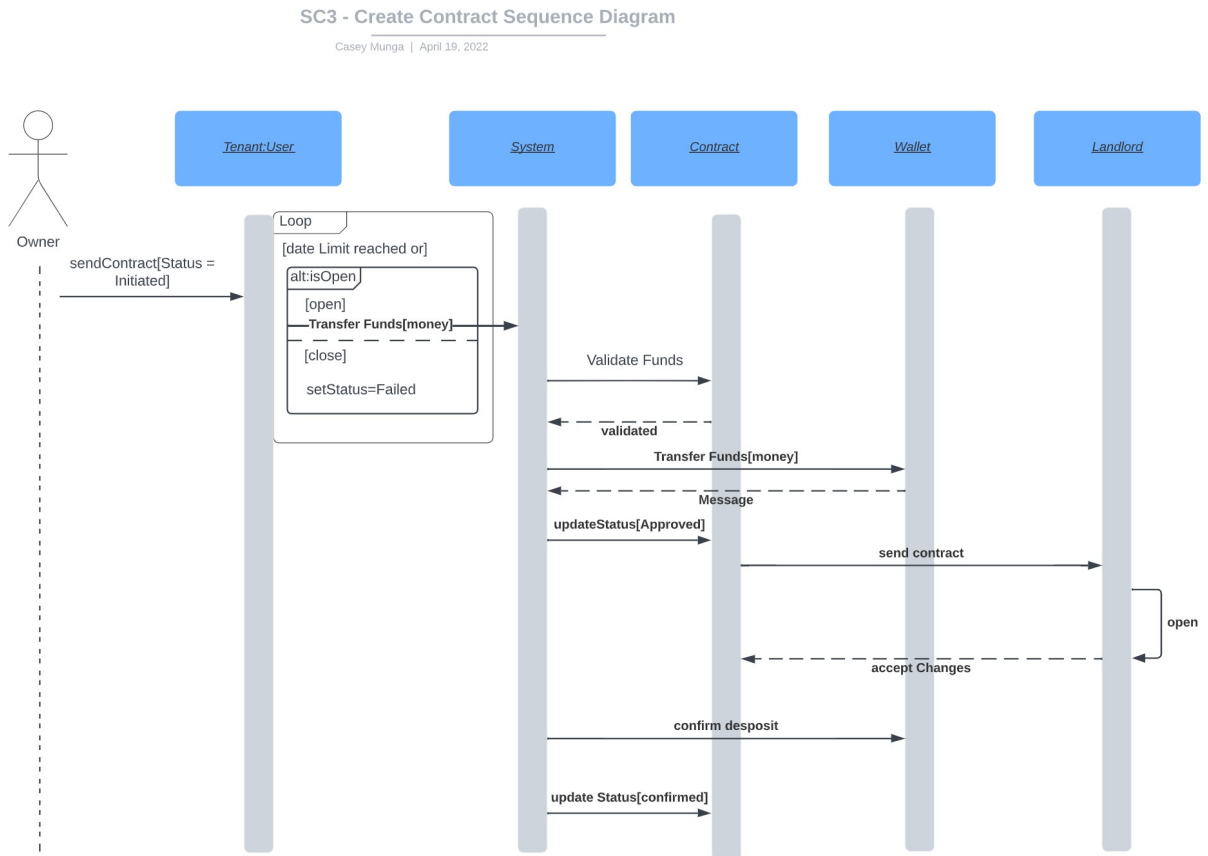


## 7.4 Create Contract

### 7.4.1 Activity Diagram

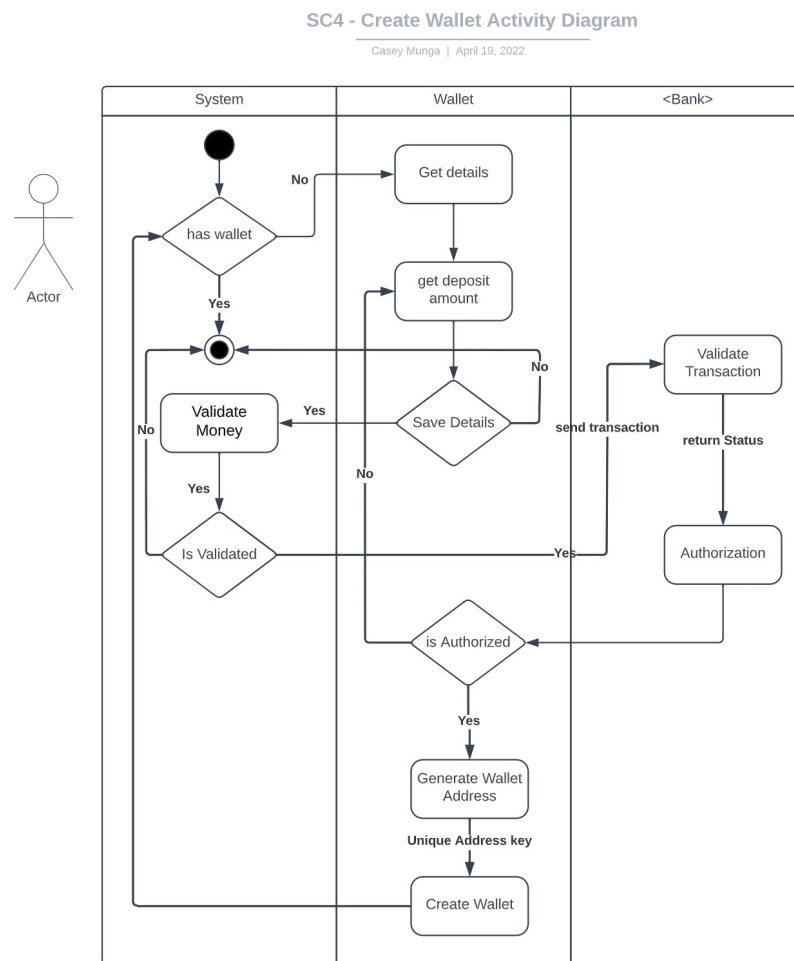


## 7.4.2 Sequence Diagram

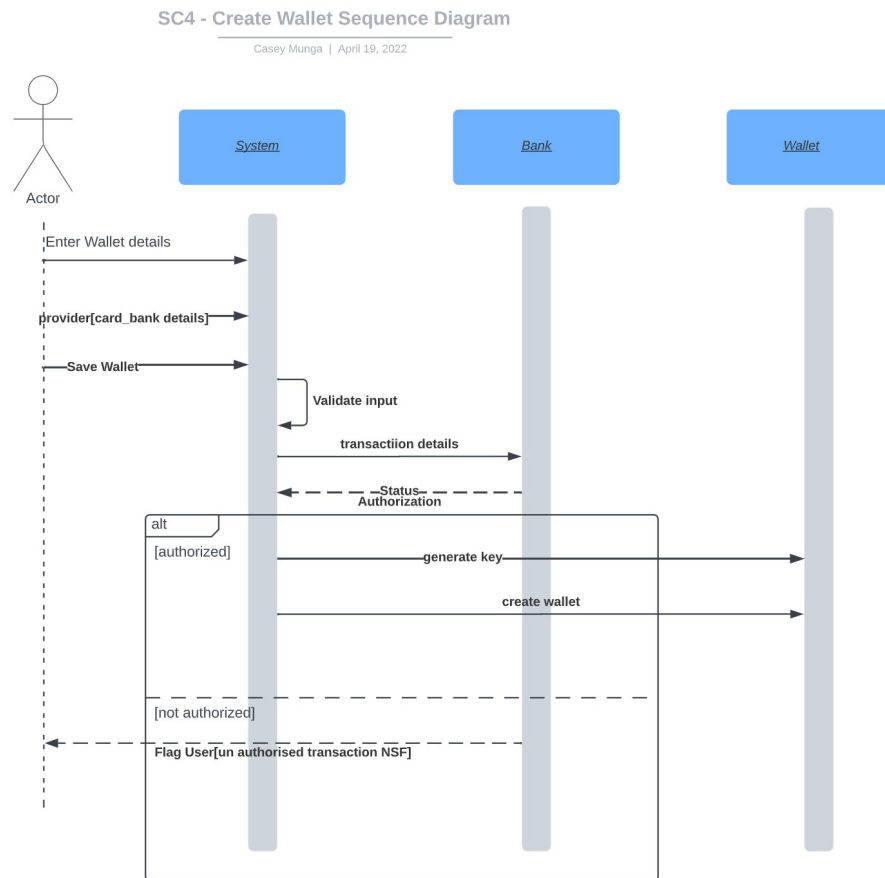


## 7.5 Create Wallet

### 7.5.1 Activity Diagram



## 7.5.2 Sequence Diagram



## 7.6 Deploy Contract

### 7.6.1 Activity Diagram

