

Software Design Document for SAYA.life

Version 1.0 approved

Prepared by Chris Case

SAYA.life

May 5, 2023

Table of Contents.....	
Revision History.....	
1. Introduction.....	
1.1. Purpose.....	
1.2. Document Conventions.....	
1.3. Intended Audience and Reading Suggestions.....	
1.4. System Overview.....	
2. Design Considerations.....	
2.1. Assumptions and dependencies.....	
2.2. General Constraints.....	
2.3. Goals and Guidelines.....	
2.4. Development Methods.....	
3. Architectural Strategies.....	
4. System Architecture.....	
4.1.	
4.2.	
5. Policies and Tactics.....	
5.1. Specific Products Used.....	
5.3. Testing the software.....	
5.5. Guidelines and conventions.....	
5.6. Protocols.....	
5.7. Maintaining the software.....	

5.8.	Interfaces.....	
5.9.	System's deliverables.....	
5.10.	Abstraction.....	
6.x	Detailed System Design.....	
6.x.1	Responsibilities.....	
6.x.2	Constraints.....	
6.x.3	Composition.....	
6.x.4	Uses/Interactions.....	
6.x.5	Resources.....	
6.x.6	Interface/Exports.....	
7.x	Detailed Lower level Component Design.....	
7.x.1	Classification.....	
7.x.2	Processing Narrative(PSPEC).....	
7.x.3	Interface Description.....	
7.x.4	Processing Detail.....	
7.x.4.1	Design Class Hierarchy.....	
7.x.4.2	Restrictions/Limitations.....	
7.x.4.3	Performance Issues.....	
7.x.4.4	Design Constraints.....	
7.x.4.5	Processing Detail For Each Operation.....	
8.	User Interface	
8.1.	Overview of User Interface.....	
8.2.	Screen Frameworks or Images.....	
8.3.	User Interface Flow Model.....	

9. Database Design	
10. Requirements Validation and Verification.....	
11. Glossary.....	
12. References.....	

Revision History

Name	Date	Reason For Changes	Version
Chris Case	10/10/2022	Rough Draft Started	0.1
Chris Case	3/20/2023	Final Draft Started	1.0
Chris Case		Final Draft Completed	1.1

1. Introduction

1.1. Purpose

The purpose of this program is to allow users to manage their water usage and pay their water bill, for both multi-tenant and single-tenant users. We will be updating the current Saya.Life website and increasing both the modularity of the site overall, but also streamlining the visual interface to make it easier for users to traverse. In addition, the web applications forecasting/data analytics will be updated to make it more readable and accessible to users with varied levels of comfort with interfacing with web based applications.

1.2. Document Conventions

This document will be written in standard MLA format.

1.3. Intended Audience and Reading Suggestion

This document is intended for developers, staff, users, testers, and documentation team, employed by or working with Saya INC, specifically as it pertains to the dashboard and subsequent web pages associated with their sensor/water usage management system.

1.4. System Overview

This software system will provide an overhaul to the current web design used by Saya.Life for their water management system. A key focus will be placed on modularity, allowing each individual user to customize the dashboard and subsequent pages, reducing unnecessary clutter or unwanted features. In addition, the data visualizations will be reworked to facilitate this increase in modularity and water consumption forecasting improved as well. The option for user's to pay their water bills will also be added. As the company has also expressed an interest in encouraging smart water consumption habits, a threshold system will also be implemented to allow users to control how much water they use.

2. Design Considerations

2.1. Assumptions and Dependencies:

- **Related software or hardware**
 - React
 - Javascript
 - Node.js
 - mySQL
 - HTML
 - CSS
- **Operating system**
 - Windows
 - MAC
- **Related Programs**
 - Visual Studios Guide
 - Figma
 - Postman
 - Excel
 - Github

2.2. General Constraints

- User must have access to internet to check their account
- Issues with sensors could cause data errors
- Some concerns around privacy as it pertains to multi-tenant users having access to single tenant users information

2.3. Goals and Guidelines

- Implement an Agile framework to ensure our design progresses smoothly
- Create a modular web page that allows users to fully customize their experience
- Allow users to pay their bills
- Create forecasting capabilities
- Ability to set thresholds to meet the forecasted metrics
- Insure the environment is a secure one as billing and the option to shut off valves will be allowed.

2.4. Development Methods

We chose to implement the Agile Scrum development method for our project implementation. Using Agile, we divide our weeks into sprints(which can be either one or two week intervals) and we create epics, which are deliverables containing subcategories of tasks needed for their completion. Each weekly meeting should only last 15 minutes by traditional Agile methodology, but our group has agree to a lot more time per meeting if things remain unclear. The Saya software team will also be asked to join in on some of our scrums to ensure the project remains correctly oriented.

3. Architectural Strategies

3.1. Use of Particular Products

- 3.1.1. Javascript and React: Our code was written in Javascript using the React library as specified by our liaison. We were able to create several modules/components that the Saya.Life design team can expand upon as they see fit or fully implement. Once our team learned how the React library function, the creation of the modules was a very streamlined process.
- 3.1.2. mySQL was used to create a functioning database to provide a backend that our React modules could interact with and simulate the pulling of data from Saya.life's inhouse database
- 3.1.3. Github was used to push all of the implemented code and allowed designers to quickly update their assigned tasks
- 3.1.4. Atlassian was used to monitor progress on the overall project and assign tasks
- 3.1.5. Discord was our primary communication for issues that arose within the team

3.2. Reuse of Existing Software Components

Beyond using the existing Saya webpage as a template our web application was built from the ground up using the React library as specified.

3.3. Feature Plans for Software Enhancement

- Fully implement the tenant view of the web application
- Create forecasting data analytics
- Connect to third party billing application
- More user testing
- Running tests to ensure application is ADA compliant

3.4. User Interface Paradigms

- 3.4.1. The user will be presented with the Login Page upon initial access
- 3.4.2. Upon a successful login the user will be taken to the Dashboard page
 - 3.4.2.1. The dashboard has a sidebar containing navigation to the follow on modules.
 - 3.4.2.2. Dashboard will display logged in user's information at top of page
 - 3.4.2.3. Dashboard will also feature notifications, quick

action bar, analytics, and calendar

- 3.4.3. Tenants page will feature a table containing information on all tenants associated with the given account
 - 3.4.3.1. Tenant page will also allow landlords to add new tenant
- 3.4.4. Billing page will allow user to access third party billing software
- 3.4.5. Analytics page will feature charts associated with account's water usage
- 3.4.6. Hardware page will provide table with all devices associated with the account
- 3.4.7. Notifications page displays all notifications associated with the account
 - 3.4.7.1. Also allows landlords to send notifications to given tenant
- 3.4.8. Settings page allows user to change account data
- 3.4.9. Logout button will return user to the login page

3.5. Hardware Interface Paradigm

- 3.5.1. Device must feature a display

3.6. External Database

For the purpose of our project a mySQL database was created to allow for full functionality, using the same fields from Saya's internal database for future implementation.

4. System Architecture

4.1. Overview

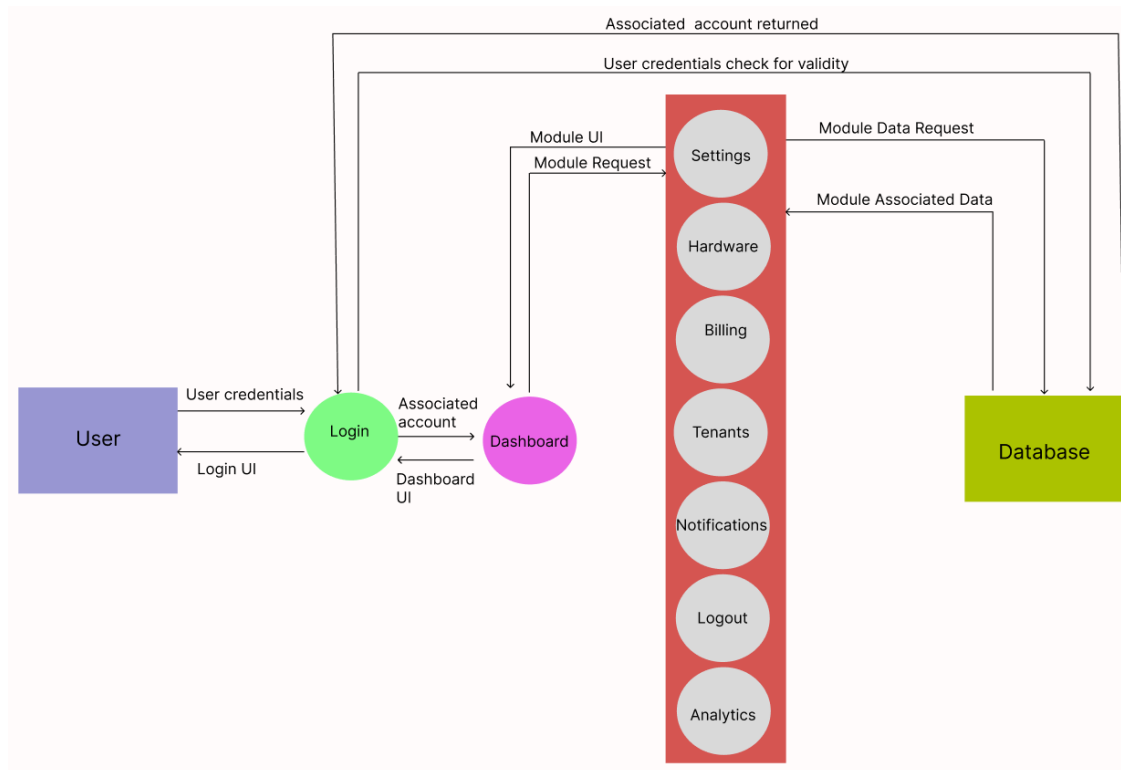


The User: the customer interacting with the application, specifically a landlord for our teams implementation

The Application: a React developed web application containing all the components associated with the application

The Database: contains all user associated data

4.2. Data Flow



The Saya web application features eight different pages

4.2.1 Login : The login page is the first page the user sees and ensures someone with the correct credentials is allowed to access the follow on pages.

4.2.2 Dashboard: The dashboard features all the follow on modules as well as displaying a calendar, notifications module, quick action bar, and analytics chart.

4.2.3 Settings: Settings page will allow users to make changes to their profile information.

4.2.4 Hardware: The hardware page will display all the devices, sensors, and gateways associated with a landlord's account as well as their tenants devices.

4.2.5 Billing: The Billing page will feature a link to the third party software that the Saya.Life application uses for handling account billing.

4.2.6 Tenants: The tenants page will have a table displaying all the tenants associated with a given landlord's account.

4.2.7 Notifications: The notifications page has a table displaying all notifications associated with an account and will allow landlords to send notifications to tenants.

4.2.8 Analytics: The analytics page displays charts and graphs associated with the water usage of an account, allowing landlords to track individual tenants' water consumption.

4.3. Software Development and Implementation

Our application is a redesign of the existing Saya web application so our first step as a group was to analyze their existing API and create wireframes to brainstorm on how we eventually wanted the site to look. The group also studied React to ensure we understood how to make full use of the tools the library offered. For the analytics portion we initially attempted a D3 library implementation, then we decided on trying to use Tableau sheets for dynamic analytic creation, before finally deciding on using Papaparse in the Javascript library, as that ultimately proved to be the most effective from an implementation standpoint. The database was created using MySQL and made use of the fields we were provided by the Saya.Life design team to create an easy porting situation when they finally received our application.

5. Policies and Tactics

5.1 Choice of which specific products used

- Visual Studio Guide
- Atlassian Jira Board (Agile framework tool)
- Github
- Discord
- Zoom
- Skype (used when communicating with Saya team)
- Canva
- Figma

5.2 Plans for ensuring requirements traceability

As the webpage is developed, each developer will create test cases for each of the functions they are implementing. Github will be used to allow developers to branch off from the main and design components as they are assigned to them.

5.3 Plans for testing the software

Our team will make use of friends and family members to provide user feedback driven testing, as well as conducting our own tests as well. We are aiming to have a wide array of testers, particularly with disparity between the age groups we pick, to ensure our webpage is intuitive to all users. The team will be testing for bugs and other issues that may arise during the development stage of web design as well.

5.4 Maintaining the Software

Upon completion of our application the Saya.Life design team will take over maintenance and further design of the web application. They will be responsible for application security and any additional features they decide are needed. Application updating will fall under their purview and they will adjust the components to meet their follow on needs.

6. Detailed System Design

6.1.1 Login.jsx

6.1.1.1 Responsibilities

The landing page once the application is accessed. Will server as as the buffer between the applications features and can only be accessed upon successfully entering credentials associated with a given account.

6.1.1.2 Constraints

A user must have created an account to be able to log in.

6.1.1.3 Composition

This component displays a field for entering a user's email and password, as well as a button that links to the forgot password page

6.1.1.4 Uses/Interaction

This component links the user to the dashboard page

6.1.1.5 Interface/Exports

React.js and CSS

6.1.2 Dashboard.jsx

6.1.2.1 Responsibilities

Accessed after a user logs in. Displays the logged in user's information and contains a sidebar with links to all additional pages in the application as well as modules for notifications, the calendar, quick action bar, and a graph from the analytics page.

6.1.2.2 Constraints

The user must have successfully logged in to access.

6.1.2.3 Composition

Displays a sidebar with links to additional pages to the left and, in the main field on the right, displays the user info, notifications, quick action bar, calendar, and a graph from analytics.

6.1.2.4 Uses/Interaction

This component links to all other components in the web application.

6.1.2.5 Interface/Exports

React.js and CSS

6.1.3 Notifications.jsx

6.1.3.1 Responsibilities

This component displays all the notifications associated with a user's tenants and allows the user to send notifications to any tenant connected to their account.

6.1.3.2 Constraints

The notifications page must be connected to the database and the field which allows landlords to send notifications to tenants must also be connected to the corresponding tenant account.

6.1.3.3 Composition

The component contains a button to allow users to send a notification, the ability to toggle notifications on/off, three fields for searching for a particular notification, and a table of all the notifications associated with a given account.

6.1.3.4 Uses/Interaction

This component allows landlords to view all notifications connected to their account and their tenant's accounts. Also allows a landlord to create and send a notification to a tenant.

6.1.3.5 Interface/Exports

React.js and CSS

6.1.4 Tenant.jsx

6.1.4.1 Responsibilities

This component displays all of the tenants associated with a given landlord's account. Also allows the landlord to add a new tenant and search for a tenant.

6.1.4.2 Constraints

This component must be connected to the database and when a landlord adds a new tenant, the corresponding tenant's *landlord_id* field in the database needs to update with the new information

6.1.4.3 Composition

At the top of this component are three fields are adding a new tenant to their account, a field for searching for a given tenant, and a list of all tenants associated with their account.

6.1.4.4 Uses/Interaction

This component allows landlords to view their tenants and add new tenants.

6.1.4.5 Interface/Exports

React.js and CSS

6.1.5 Billing.jsx

6.1.6.1 Responsibilities

This component will display the current billing information for tenants connected to this account, notifications table for any notifications associated with billing, and a search field to find a given tenant's billing information.

6.1.6.2 Constraints

This component must be connected to the database and be correctly joined in the database.

6.1.6.3 Composition

This component displays a table for notifications associated with billing at the top of the page, a field for search for a given tenant's bill, and a table listing all the billing information for a landlord's tenants.

6.1.6.4 Uses/Interaction

This component shows a landlord any billing related notifications and allows a landlord to view the billing information for associated tenants.

6.1.6.5 Interface/Exports

React.js and CSS

6.1.7 Settings.jsx

6.1.7.1 Responsibilities

This component allows a user to change both their account information and the personal information connected to their account.

6.1.7.2 Constraints

This component must be connected to the database and have the necessary queries to update the database correctly.

6.1.7.3 Composition

Contains twelve different fields that update a user's personal information and the information connected to their account, such as password, name, account name, phone number, etc.

6.1.7.4 Uses/Interaction

A user is able to change their account information by entering new values in the fields and selecting the save button at the bottom of the page. This will update the corresponding fields in the database.

6.1.7.5 Interface/Exports

React.js and CSS

6.1.8 Analytics.jsx

6.1.8.1 Responsibilities

This component displays graphs and charts tracking water usage for all the tenants connected to the landlord's account.

6.1.8.2 Constraints

This component must be connected to the sensor data in the database so provide accurate reports.

6.1.8.3 Composition

This component displays different charts and graphs showing how a tenant is using water.

6.1.8.4 Uses/Interaction

The landlord is able to view the charts and graphs with water usage, flow rate, and other metrics.

6.1.8.5 Interface/Exports

React.js and CSS

6.1.9 forgotpassword.jsx

6.1.8.1 Responsibilities

This component will direct the user to a page featuring a field for them to enter their email. Upon clicking the send button, an email will be sent to the corresponding email to allow a user to change their password.

6.1.8.2 Constraints

There must be an automated email attached to the send button of this component that sends a preformatted email, with attached link, to the user designated email.

6.1.8.3 Composition

This component displays a field for the user to enter their email along with a send button to notify system to send the form email to the corresponding address.

6.1.8.4 Uses/Interaction

The user will enter their email in the given field and interact with the send button.

6.1.8.5 Interface/Exports

React.js and CSS

7. Detailed Lower level Component Design

7.1 Sidebar.jsx

7.1.1 Classification

Displays links to notifications, settings, tenants, analytics, billing, and hardware pages.

7.1.2 Processing Narrative (PSPEC)

Upon interaction with each link the user will be brought to the specified page.

7.1.3 Interface Description

An interactive bar spanning the left side of the dashboard page, containing clickable links to the follow on pages of the web application.

7.2 QuickActions.jsx

7.1.1 Classification

Displays selectable icons including set threshold, send email, add new tenant, and export report.

7.1.2 Processing Narrative (PSPEC)

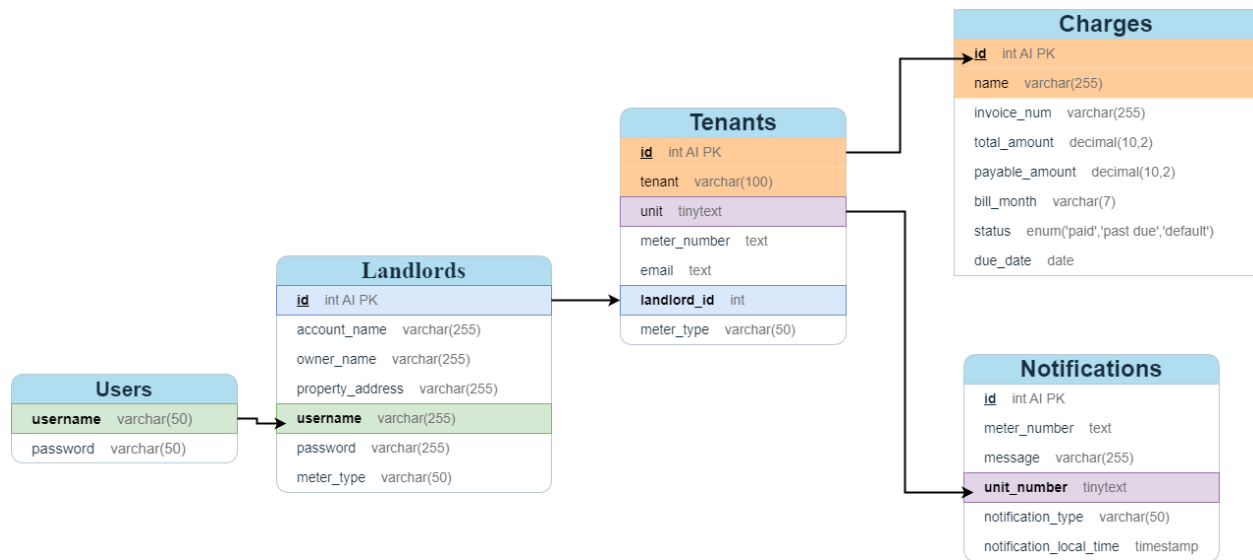
The user will select one of the above mentioned buttons to set their water usage threshold, send an email to a tenant associated with their account, add new tenant information, and produce a report of water usage for a landlord's tenants.

7.1.3 Interface Description

A clear bar located in the top right corner of the dashboard page, containing the four above mentioned buttons.

8. Database Design

The database for our web application was created in MySQL and was updated through the IDE our team used (Visual Studio Guide). The database contained fields provided to the team by Saya.life to ensure ease of portability when the Saya design team takes over the application. By utilizing the given fields, our team was able to create analytics using customer data to ensure accuracy and realistic data visualization. Future implementations, when connected to the database will have the ability for real time analysis and also forecasting of future water usage.



```
const [tenant, settenant] = useState("");
const [unit, setunit] = useState("");
const [meter_number, setMeterNumber] = useState("");
const [email, setEmail] = useState("");
const [tenantlist, settenantlist] = useState([]);

const addtenant = () => {
  Axios.post("http://localhost:3001/create", {
    tenant: tenant,
    unit: unit,
    meter_number: meter_number,
    email: email,
  }).then(() => {
    settenantlist([
      ...tenantlist,
      {
        tenant: tenant,
        unit: unit,
        meter_number: meter_number,
        email: email,
      },
    ]);
  });
}
```

Information

Table: test_users

Columns:

id	int PK
email	varchar(255)
password	varchar(255)
country_code	varchar(255)
phone_number	varchar(255)
account_name	varchar(255)
invoice_name_prefix	varchar(255)
first_name	varchar(255)
last_name	varchar(255)

Object Info Session

9. User Interface

9.1 Overview of User Interface

The application will have two views depending on the login credentials of the user, a landlord view and a tenant view (TBD). The landlord view will have added functionality not present in the tenant view and users will not be able to access corresponding views.

9.1.1 The user will be directed to the login page upon web application initialization

9.1.2 The user will be directed to the dashboard page upon successful login

9.1.2.1 The login page will feature the user's information at the top of the page.

9.1.2.2 A sidebar containing links to follow on pages will be on the left.

9.1.2.3 The login page will feature a notification table in the top right corner.

9.1.2.4 The quick action bar will be located at the top left.

9.1.2.4 There will be a calendar under the quick action bar.

9.1.2.5 An analytics graph will be at the bottom of the page

9.1.3 The tenant page shall display a table listing all the tenants associated with a Landlord.

9.1.3.1 Fields at the top of the page will be for adding a new tenant.

9.1.3.2 A search field will be present for directing to a given tenant.

9.1.4 The billing page will feature a notification table with all billing notifications.

9.1.4.1 A search field will be utilized for locating a given bill.

9.1.4.2 All tenants associated with the landlord will have their billing information displayed.

9.1.5 The analytics page will feature charts and graphs displaying tenant's water usage.

9.1.6 The hardware page shall display all the devices and sensors connected to

tenants under a landlord.

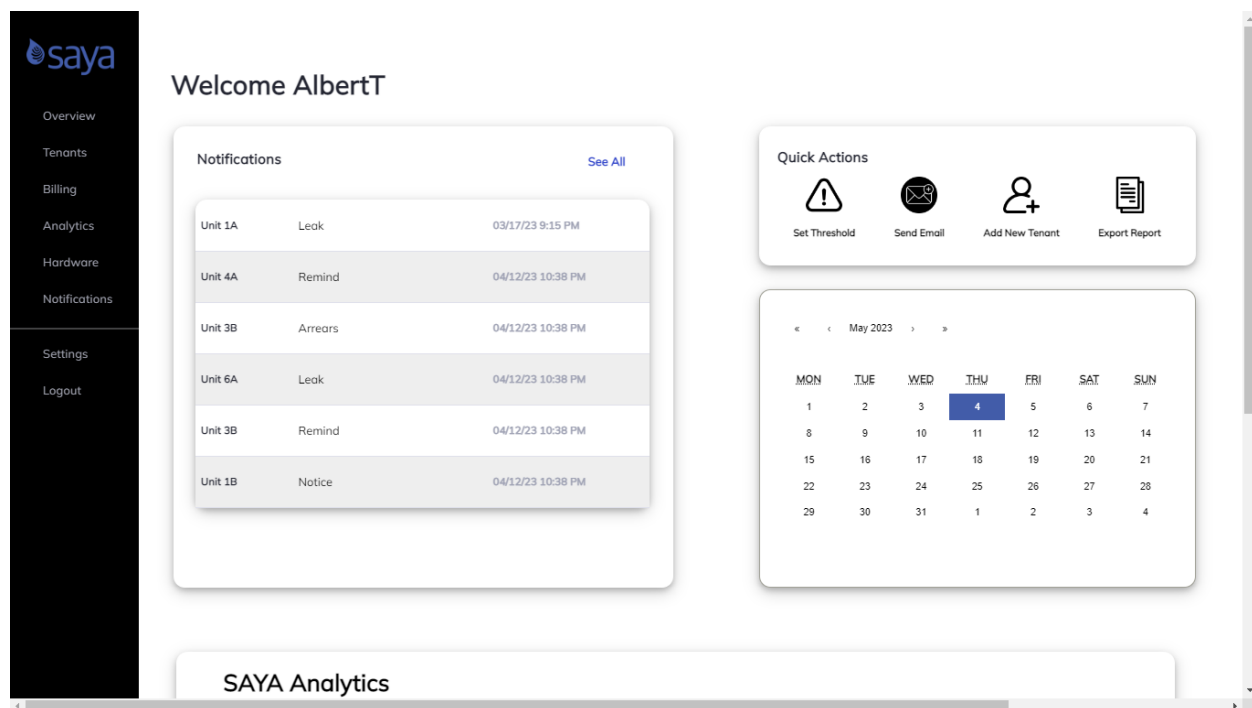
9.1.7 The notifications page shall display a table with all notifications from tenants connected to a landlord's account.


9.1.7.1 A create new notifications button is displayed at the top of page

9.1.7.2 Three fields are present for searching for a given notification

9.1.7.3 A toggle is present to turn off notifications

9.2 Screen Frameworks or Images





Overview

Tenants

Billing

Analytics

Hardware

Notifications

Settings

Logout

Notification Mangager | Status: On

Create New Notification

Notifications Enabled? ☒

Notification History

Search (Enter minimum 3 letters)

WH67890


From Date

To Date

End Date

Q

Meter Number	Message	Unit #	Notification	Time
WH123245	Leak detected in basement	1A	Leak	03/17/23 9:15 PM
WH428y497869	Water cut off notice	4A	Remind	04/12/23 10:38 PM
WH1234356	Arrears	3B	Arrears	04/12/23 10:38 PM
WH1232468	The water pipes are leaking	6A	Leak	04/12/23 10:38 PM
WH1234356	Renewal reminder	3B	Remind	04/12/23 10:38 PM
WH67890	Excessive water	1B	Notice	04/12/23 10:38 PM
WH9798748	Renewal reminder	7A	Remind	04/12/23 10:38 PM
WH456789	Arrears	3C	Arrears	04/12/23 10:38 PM
WH123250	pipes are leaking.	2C	Leak	04/12/23 10:38 PM
WH234567	Water cut off notice	4B	Remind	04/12/23 10:38 PM



Overview

Tenants

Billing

Analytics

Hardware

Notifications

Settings

Logout

Tenants Overview

Add New Tenant

Tenant: Full Name

Unit: ex. 24

Meter Number: ex. WHG223F

Email: Valid Email

















Add New Tenant

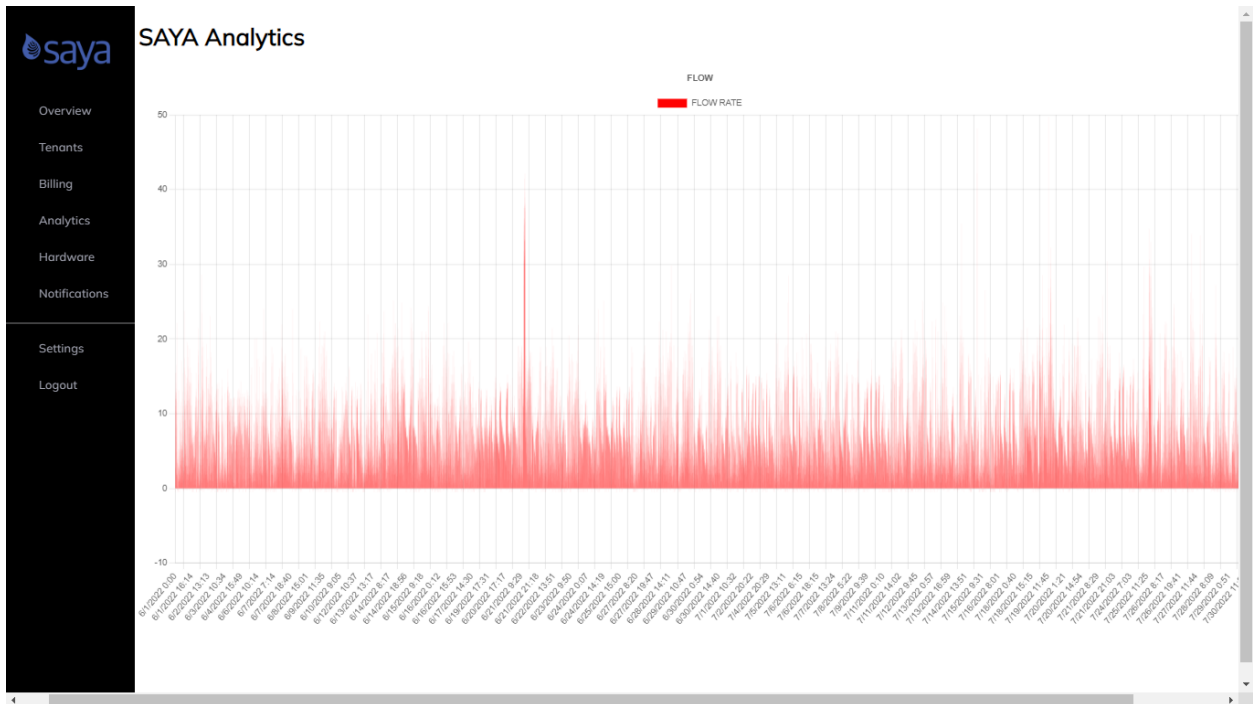
View Tenants

Search (Minimum 3 Letters)

Search by Name or Invoice Number

Q

Tenant Name	Unit	Meter Number	Email	Options
John Smith	1A	WH123245	johnsmith@gmail.com	 
Sarah Johnson	1B	WH67890	sarahjohnson@gmail.com	 
Dwayne Johnson	7A	WH9798748	TheRock@gmail.com	 
Steve Austin	4A	WH428y497869	Steve316@gmail.com	 
Chris Case	3A	WH1234556	Chriscase@gmail.com	 
Michael Davis	1C	WH908976	michaeldavis@gmail.com	 
Karen Lee	2A	WH746830	karenlee@gmail.com	 
John Doe	3B	WH1234356	johndoe@example.com	 



Billing Overview

Notifications [See All](#)

Unit 1A	Leak	03/17/23 9:15 PM
Unit 4A	Remind	04/12/23 10:38 PM
Unit 3B	Arrears	04/12/23 10:38 PM

Billing History

Search (Minimum 3 Letters)

Search by Name or Invoice Number

Name	Invoice Number	Total Amount	Payable Amount	Bill Month	Status	Due Date
John Smith	INV49470	103	95.77	2022-03	past due	2023-05-12T07:00:00.000Z

- Overview
- Tenants
- Billing
- Analytics
- Hardware
- Notifications
- Settings
- Logout

Notification Manager | Status: On

Create New Notification
Notifications Enabled? ☒

Notification History

New Notification

Notification Type

Unit Number

Meter Number

Notification Preview

Attention [Tenant], This is a test notification!

Send

Unit Number	Notification Type	Category	Action	Time
WH1234356	Renewal reminder	3B	Remind	04/12/23 10:38 PM
WH67890	Excessive water	1B	Notice	04/12/23 10:38 PM
WH9798748	Renewal reminder	7A	Remind	04/12/23 10:38 PM
WH456789	Arrears	3C	Arrears	04/12/23 10:38 PM
WH123250	pipes are leaking.	2C	Leak	04/12/23 10:38 PM
WH234567	Water cut off notice	4B	Remind	04/12/23 10:38 PM

- Overview
- Tenants
- Billing
- Analytics
- Hardware
- Notifications
- Settings
- Logout

Water Sensors & Hardware

Michael Chen

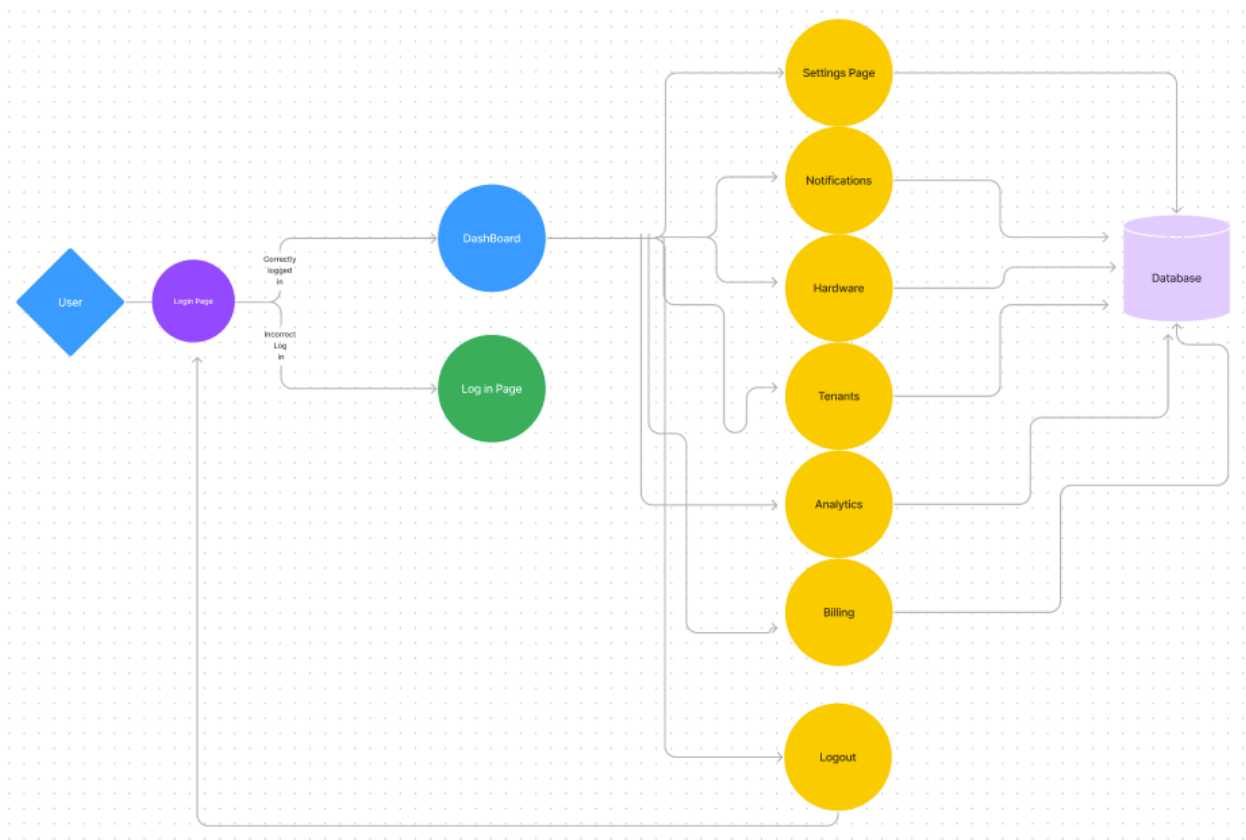
Meter type: gateway

Tenant Name	Meter Number	Meter Type
John Smith	WH123245	1 inch ultrasonic meter
Steve Austin	WH428y497869	1 inch ultrasonic meter
John Doe	WH1234356	1 inch ultrasonic meter
Lillian Kim	WH1232468	1 inch ultrasonic meter
David Kim	WH123249	1 inch ultrasonic meter
Peter Wang	WH123251	1 inch ultrasonic meter
Kevin Lee	WH1232423	1 inch ultrasonic meter
Rachel Smith	WH123260	1 inch ultrasonic meter
Olivia Lee	WH1232462	1 inch ultrasonic meter
Peter Johnson	WH1232465	1 inch ultrasonic meter

Maria Garcia

Meter type: gateway

9.3 User Interface Flow Model



10. Requirements Validation and Verification

10.1 Login

10.1.1	The system shall window with two fields for username and password.	
10.1.2	The system shall display a forgot password button.	

10.2 Dashboard

10.2.1	The system shall display the user's information at the top of the page.	
10.2.2	The system shall display a sidebar with links to follow on pages.	
10.2.3	The system shall display a notification window.	
10.2.4	The system shall display a quick action bar.	
10.2.5	The system shall display a calendar.	
10.2.6	The system shall display a graph from analytics.	

10.3 Settings

10.3.1	The system shall display a field to change account name.	
10.3.2	The system shall display a field for updating the country code of a user's phone number.	
10.3.3	The system shall display a field for changing name prefix.	
10.3.4	The system shall display a field for changing the user's email.	
10.3.5	The system shall display a field for changing a user's phone number.	
10.3.6	The system shall display a field for displaying the account	

	name.	
--	-------	--

10.4 Billing

10.4.1	The system shall display a window for notifications tied to a tenant concerned with focussed on billing.	
10.4.2	The system shall display a field for searching for a given tenant's billing information.	
10.4.3	The system shall display a table with all the billing information for a landlord's tenants.	

10.5 Notifications

10.5.1	The system shall display a button for creating a new notification.	
10.5.2	The system shall display three fields for searching for a particular notification.	
10.5.3	The system shall display a table with all notifications associated with the tenants under a given landlord.	
10.5.4	The system shall display a toggle for switching off notifications.	

10.6 Tenant

10.6.1	The system shall allow a landlord to add a new tenant.	
10.6.2	The system shall display fields for adding a new tenant.	
10.6.3	The system shall display a field to allow for searching for a given tenant.	
10.6.4	The system shall display a table of the tenants associated with a landlord's account.	

10.7 Analytics

10.7.1	The system shall display graphs and charts documenting water usage for a landlord's tenants.	
--------	--	--

10.8 Hardware

10.8.1	The system shall display a table detailing all the sensors, gateways, etc connected to the tenant's account under a landlord.	
--------	---	--

10.9 Logout

10.9.1	The system shall redirect the user to the login page.	
--------	---	--

11. References