Avinash Kola

[kola@umd.edu](mailto:kola@umd.edu)

UMD directory ID: kola

**Household Chores Distributor**

**Application Design Document**

**Table of Contents**

[Section 1 - Project Description](#_yr1n1w74g294)

[1.1 Project](#_mpniepv3sji5)

[1.2 Description](#_kxtdscgtj3ad)

[1.3 Revision History](#_x1jfzxbgcdpj)

[Section 2 - Overview](#_v0z2li3lbhgp)

[2.1 Purpose](#_wkcniuopwsk)

[2.2 Scope](#_40qz41btxs84)

[2.3 Requirements](#_b5joi8kutkvr)

[2.3.1 Estimates](#_p6lz0c1jpkfa)

[2.3.2 Traceability Matrix](#_khayjwa8jo2y)

2.4 Technology Stack

[Section 3 - System Architecture](#_yeyfpufiww8s)

3.1 System Software Architecture

3.2 Sequence Diagram

3.3 Security

3.4 Logging for Security Audits

Section 4 - System Software Components

Section 5 - Application Data Components

Section 6 - Application Software UI Components

Section 7 - Glossary

Section 8 - References

[**Section 1 - Project Description**](#_yr1n1w74g294)

* 1. **Project**

Household Chores Distributor

* 1. **Description**

The Household Chores Distributor is a user-friendly web application aims to provides a user-centric platform that allows individuals to independently manage their household tasks and preferences, promoting autonomy. It will provide an organized platform for assigning, tracking, and completing various household tasks, ensuring that everyone successfully shares responsibility.

This system is designed to share the household work among family or roommates will result in efficient chores management and equal and fair task distribution with Improved communication through E-Mail notification updates. Chores Accountability and reports include stats like history, completions, and overdue tasks will improve the transparency among the family or roommates.

## Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Comment** | **Author** |
| 06/09/2023 | Application Requirement Specification, Version 1 | Avinash Kola |
| 14/09/2023 | Application Requirement Specification, Version 2 | Avinash Kola |
| 15/09/2023 | Software Design Document | Avinash Kola |
| 10/23/2023 | Application Requirement Specification, Version 3 | Avinash Kola |
| 10/23/2023 | Software Design Document, Version 2 | Avinash Kola |

**Section 2 – Overview**

**2.1 Purpose**

The purpose of this document is to outline the design of the Household Chores Distributor system, providing a comprehensive understanding of its architecture and data structures.

**2.2 Scope**

This document covers the system's design, including its components, modules, and data schema and it defines the scope of the project in terms of functionality and design.

**2.3 Requirements**

**2.3.1 Estimates**

|  |  |  |
| --- | --- | --- |
| # | Description | Hrs. Est. |
| 1 | System Architecture (Section 3) | 5 |
| 2 | User Interface Design (Section 3) | 5 |
| 3 | Database Design (Section 4) | 5 |
| 4 | User Management Module | 5 |
| 5 | Chore Management Module | 5 |
| 6 | Chore Assignment and Tracking Module | 10 |
| 7 | Notification System Module | 5 |
|  | TOTAL | 40 |

2.3.2 Traceability Matrix

|  |  |
| --- | --- |
| **SDD Module** | **SRS Requirement** |
| Section 4- User Management | 3.2.1 User Management, 3.2.7 Manage User |
| Section 4- Chore Management | 2.2.3 Chore Management |
| Section 4- Notifications | 2.2.3 Chore Management |

**2.4 Technology Stack**

Frontend: React.js

Backend: Python (Django framework)

**Section 3 – System Architecture**

**3.1 System Software Architecture**

The Household Chores Distributor system follows a three-tier architecture, maintaining relations between User interface (UI), Application Services (Logic), and Database.

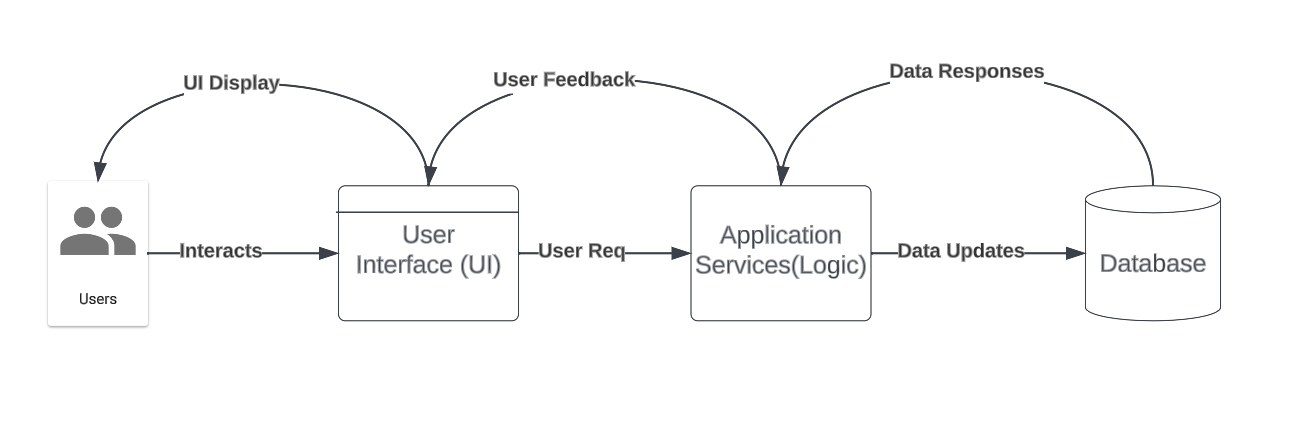


Figure – System Software Architecture

The Household Chores Distributor's system architecture is intended to give users a user-centric platform for managing household activities and preferences. It comprises multiple fundamental components that work together to provide the desired functionality.

User Interface (UI): This component represents the web-based user interface with which users interact. It has a variety of pages and layouts for managing chores, preferences, and user profiles.

The application Services (Logic): This component consists of the business logic and rules that control chore assignment, priority management, duty completion, and chore exchange. It serves as a bridge between the user interface and the data storage.

Data Storage: This component maintains all essential data, such as user profiles, chore details, customization choices, and information about household members.

**Relations:**

The User Interface (UI) component interacts with the Application Logic to request and display chore lists, assignment forms, and customization options.

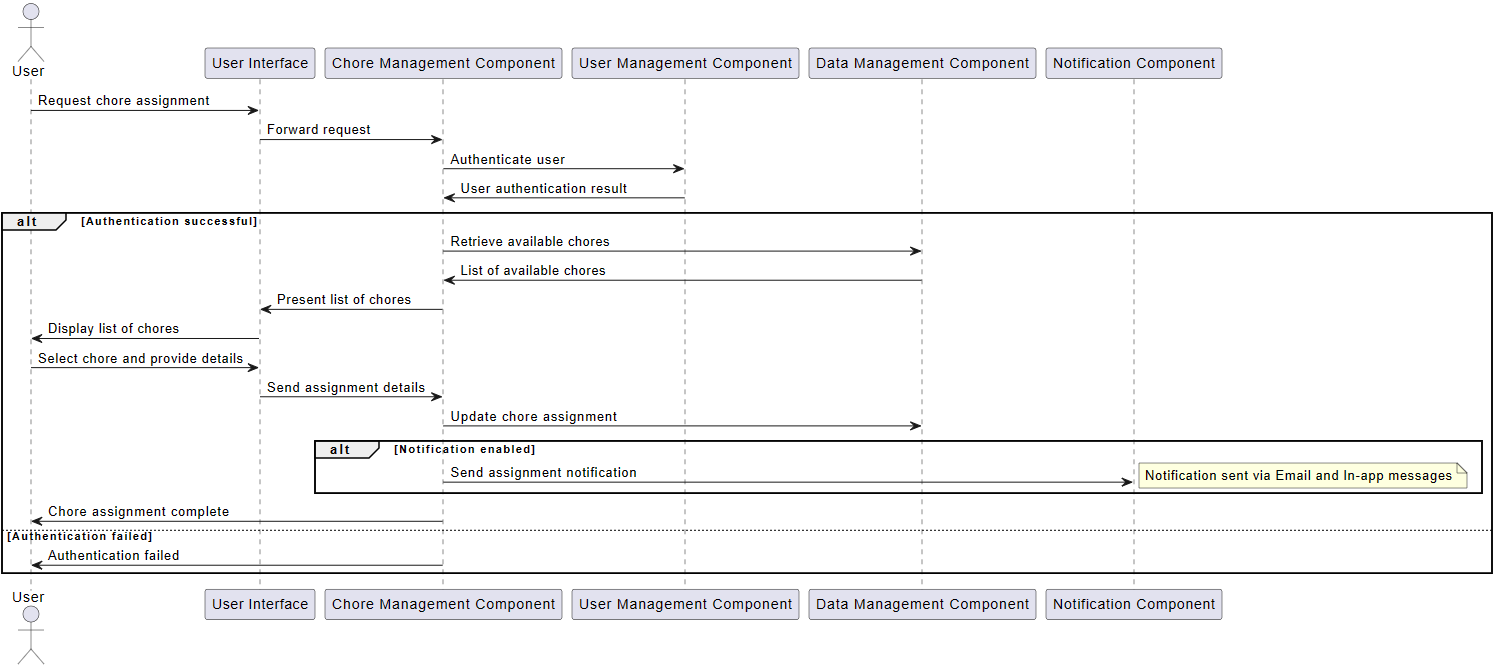
The Application Logic component communicates with the Data Storage component to retrieve and update user profiles, chore details, and customization preferences.

**System Flow:**

When a user logs in, the User Interface (UI) component sends a login request to the Application Logic and upon successful login, the UI displays the user's personalized dashboard, which includes assigned chores, priorities, and due dates.

Users can interact with the UI to create chores, set priorities, view chore lists, mark chores as completed, and delete and these UI actions trigger corresponding processes in the Application Logic, which update data in the Data Storage component.

**3.2 Sequence Diagram**

****

The sequence diagram provided above provides a concise yet informative overview of how the Household Chores Distributor system assigns chores. It shows how essential system components manage their interactions, highlighting the smooth flow of work that enables users to assign chores effectively. The diagram shows how the system manages user authentication, locates available duties, and displays them to the user, starting with the user's request through the user interface. Also, the Chore Management Component, which ensures data integrity and increases the possibility of further interactions like notifying the assigned user, helps the whole purpose of the project.

**3.3 Security**

It is crucial for the Household Chores Distributor application to prioritize security measures to protect user data and maintain system integrity. To mitigate potential vulnerabilities, the following security measures have been implemented:

Input Validation and Parameterized Queries: The application utilizes thorough input validation and parameterized queries to prevent SQL Injection. All user inputs that interact with the database are sanitized to avoid any unauthorized SQL queries from being executed. This security measure ensures that the application remains resilient against such attacks.

Cross-Site Scripting (XSS) attacks, user-generated content is automatically escaped and sanitized before being rendered on web pages. This ensures that any malicious scripts injected into the application will not be executed and user data remains secure.

Authentication and Authorization Checks: The application enforces strong authentication and authorization controls. Users must authenticate before accessing restricted areas, and proper authorization checks are in place to ensure that only authorized users can access specific functionalities. Decorators such as @login\_required and @permission\_required are utilized to maintain these controls.

Rate Limiting and Brute-Force Protection: To prevent brute-force attacks on user accounts, the application implements rate limiting and brute-force protection mechanisms. These mechanisms restrict the number of login attempts within a specific timeframe, mitigating the risk of unauthorized access through repeated attempts.

Email Account Verification: The application ensures the integrity of the email verification process. During registration, only valid and legitimate email addresses are accepted. Users are required to confirm their email addresses by clicking on a verification link sent to their provided email addresses before their accounts are activated.

Account Flooding Prevention: To mitigate the risk of account flooding, the application implements rate limiting, restricting the number of registration attempts from a single source or IP address within a specific time frame. This measure prevents the overload of the system by limiting the creation of numerous accounts in a short period.

These security measures work together to provide a robust and multi-layered security infrastructure, protecting user data and the application from potential threats.

**3.4 Logging for Security Audits**

In addition to the security measures listed in section 3.3.2, the Household Chores Distributor application has robust logging capabilities that are crucial for security auditing. Logging is essential for tracking and analyzing system activities, identifying and responding to security incidents, and ensuring compliance with security requirements. Here's how logging is integrated into the application:

Security Event Logging: The application logs all significant security-related events, such as login attempts, authentication and authorization activities, and detected security threats. These logs provide a comprehensive record of security-related actions taken by both users and the system itself.

Log Retention: The application retains logs for an extended period, ensuring historical data is available for security analysis and compliance audits. The retention policy complies with industry best practices and any legal or regulatory requirements.

Log Encryption: Logs are encrypted to protect sensitive information within the log entries, providing added security and preventing unauthorized access to log data.

Access Controls: Log access is restricted to authorized personnel only, preventing unauthorized individuals from viewing or modifying log entries.

Alerting and Monitoring: The application is configured to generate alerts or notifications for specific security events, allowing for real-time monitoring and immediate response to potential security incidents.

**Section 4 – Application Software Components**

**1. User Management Component**: This component is in control of all aspects of user accounts and authentication. it includes Modules for user registration and, login.

**2. Chore Management Component**: The chore management component controls tasks such as chore creation, assignment, and tracking. It has modules to create chore lists, assigning chores to users, prioritizing chores, delete chore and track chore completion status. This component also handles requests for chore exchanges.

**3. Notification Component**: Notifications are an important part of the system since they keep users up to date on chore assignments. The notification component manages notifications delivered through email.

**4. Data Management Component** used for System data storage and retrieval data. It has modules that manage user preferences, chore information, and chore assignments and It ensures data security and integrity.

**5. User Interface (UI) Component**: The graphical user interface elements that users interact with represent the UI component. Users may easily carry out activities and get information thanks to its web pages, forms, and dashboards because it focuses on offering a simple and user-friendly interface.

Together, these software components form a complete efficient system. Every component plays a specific role in the overall functionality of the system and the user experience. Users are able to quickly and independently handle household responsibilities within the program because of seamless communication and data flow between these components.

**Section 5 – Application Data Components**

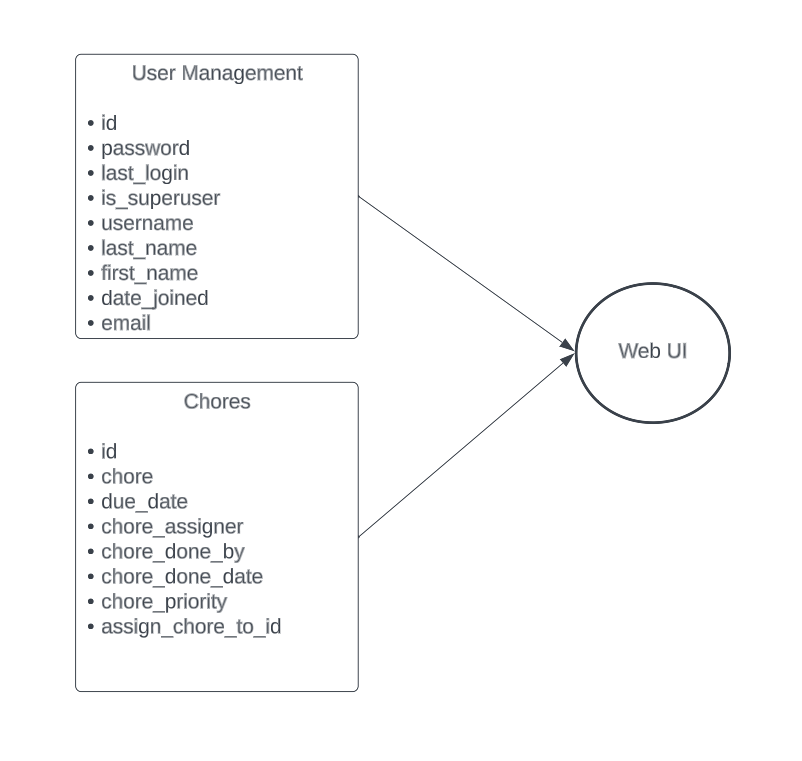
****

Figure – System Data Architecture

1. **User Management**

This component stores information about registered users of the Household Chores Distributor application.

|  |  |  |
| --- | --- | --- |
| **Data Item** | **Type** | **Description** |
| User ID | Integer | Unique identifier for users |
| Username | Text | User's username |
| Password | Text | User's password (hashed and securely stored) |
| Email | Text | User's email address |
| Is superuser | Text | Type of user (e.g., family member, roommate) |
| Date Joined | Text | Date when the user account was created |
| Last Login | Date | Shows the user's last login information |

1. **Chores**

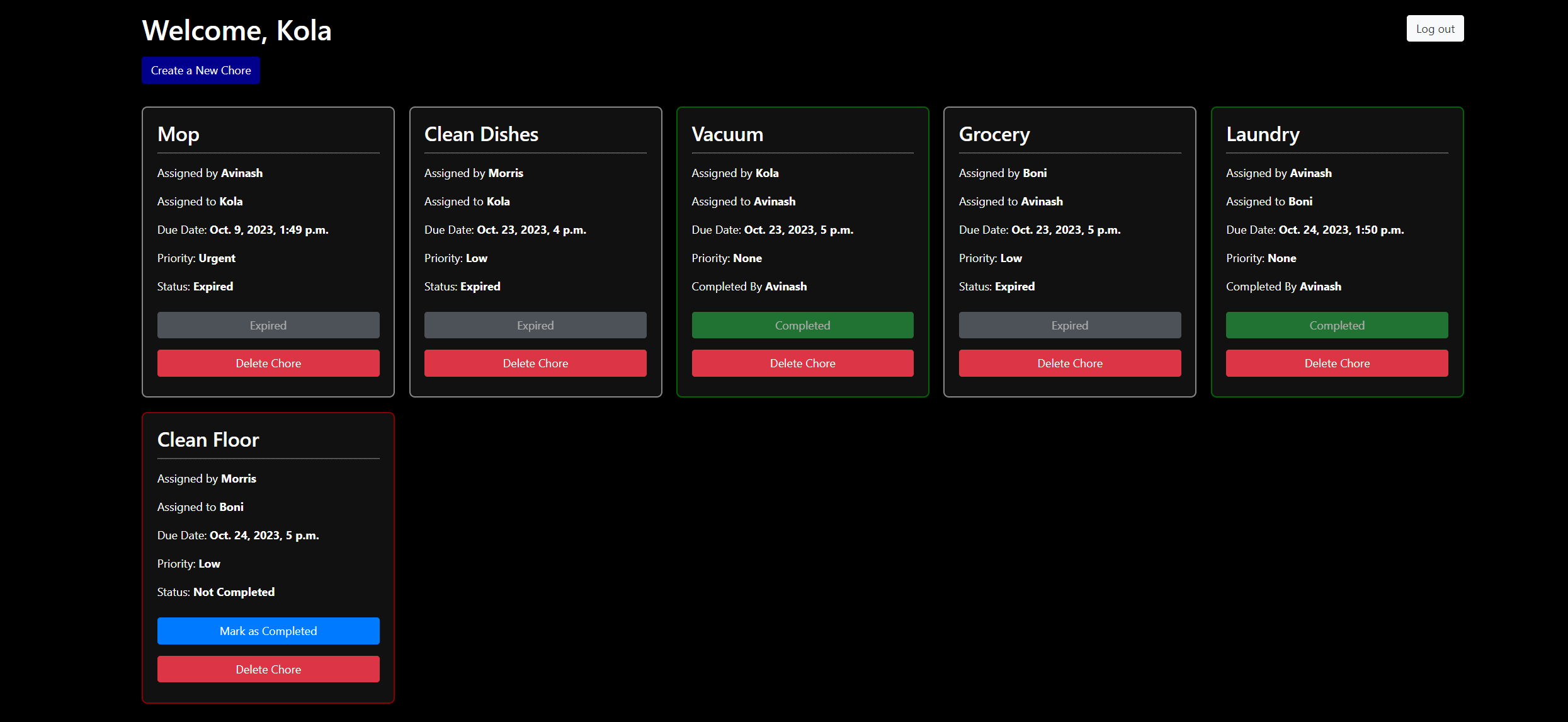
This component stores information about household chores.

|  |  |  |
| --- | --- | --- |
| **Data Item** | **Type** | **Description** |
| Chore ID | Integer | Unique identifier for chores |
| Assigned date | Date | Assigned date of the Chore |
| Chore Priority | Category | Priority level of chore (low, important, urgent) |
| Due Date | Date | Due date for completing the chore |
| Chore Assigner | Text | Name of the user who assigned the chore |
| Chore done by | Text | Name of the user who completed the chore |
| Chore done Date | Date | Date when the chore was completed |
| Status | Text | Status of the chore (assigned, completed, overdue) |
| Assigned To | Integer | The user to whom the chore is assigned |

**Section 6 – Application Software UI Components**

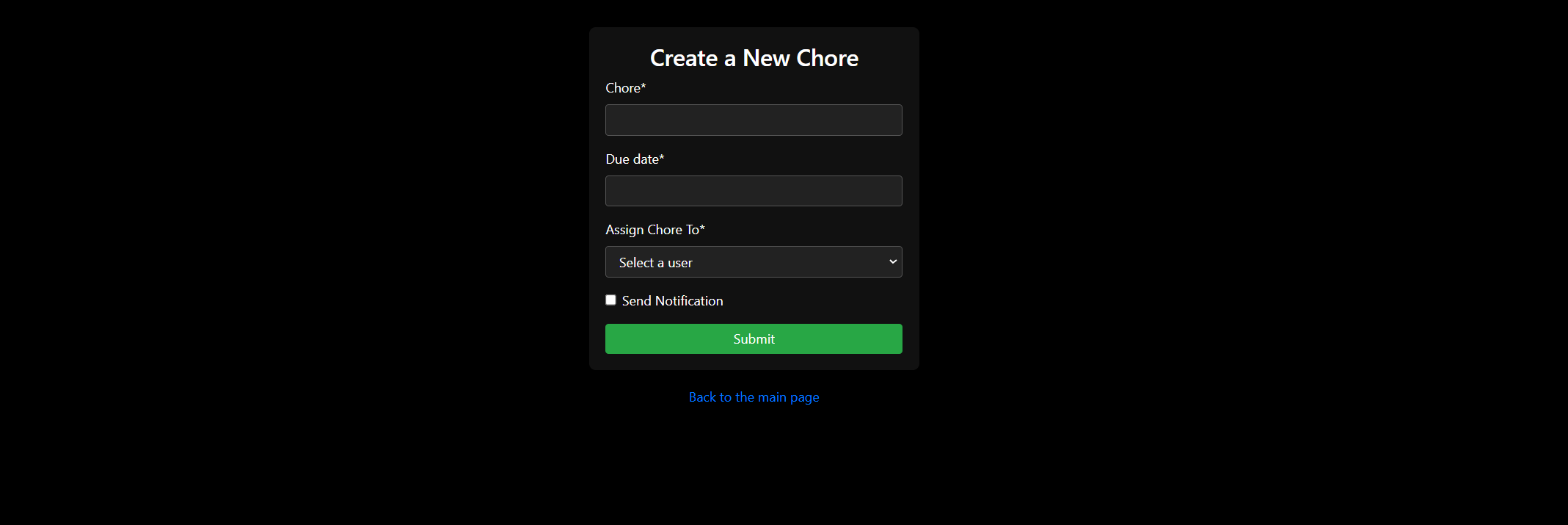
1. **Dashboard**

The dashboard will offer a simple and intuitive view of task assignments, their priority, and due dates. It will be the primary user interface component and provide a personalized view of the user's tasks and responsibilities. The tasks will be displayed in a grid box, with incomplete ones highlighted in red, completed ones in green, and expired ones in grey. The purpose of this page is to give users quick and easy access to their tasks, so they can manage and complete them efficiently.



1. **Create Chore Page**

On this page, users can create and assign tasks to themselves or other family members. To create a task, users simply need to name it, set a due date, and assign jobs as necessary. This page streamlines the process of household task management and facilitates collaboration among household members.



1. **Chore Priority Settings:**

This feature allows users to check the priority level (low, required, or urgent) assigned to each chore. It helps to categorize and manage tasks based on their importance. This option is designed to aid users in organizing their chores and identifying tasks that need immediate attention.

1. **Chore List Page:**

This page provides a detailed list of all the chores available within the household. Users can review, select, and complete or delete tasks from this list. By offering visibility into all the chores, this page makes it easier to assign and manage tasks effectively.

1. **Chore Completion Option:**

Users can use this option to mark the chores assigned to them as completed. The feature offers different options to confirm task completion and update the status. Once the user selects the "mark as complete" option, the button will turn green, indicating that the task has been completed. This page helps users track and manage chore progress by keeping them informed about the tasks that have been finished.

1. **Chore Delete Option:**

Users can use this option to delete the chores created by them.

1. **Status Option:**

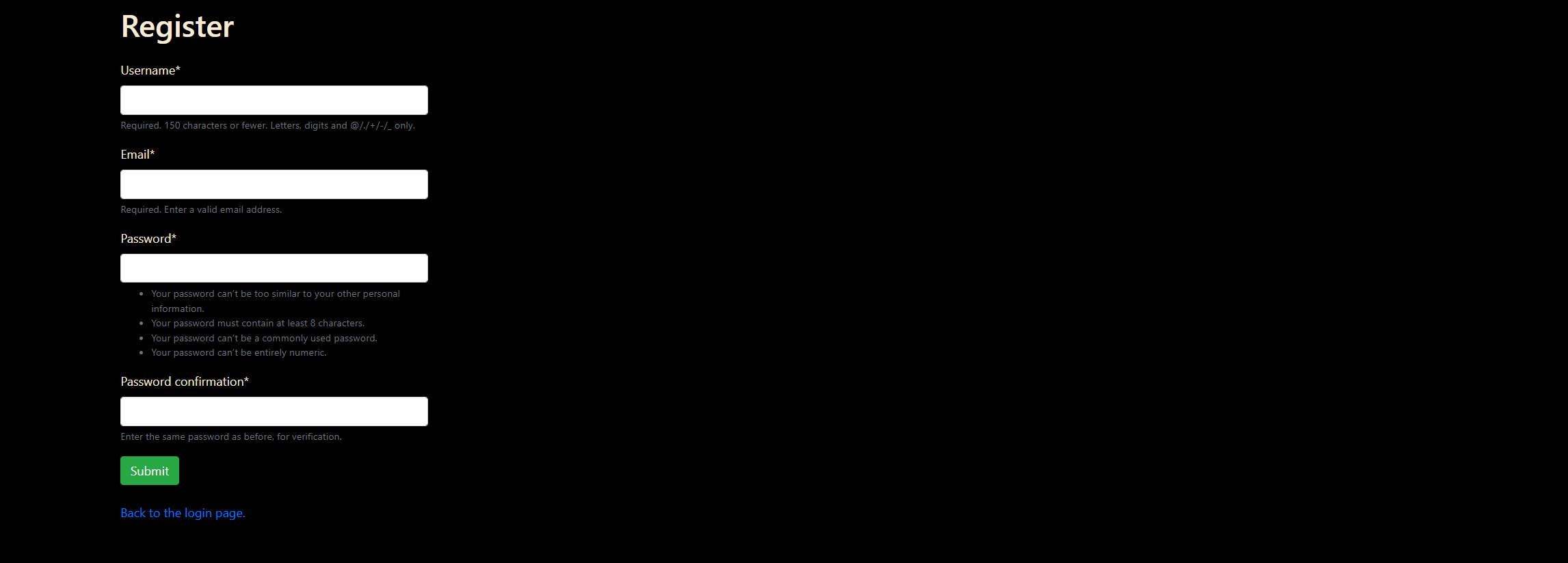
Users can view their chore status as complete, incomplete, or expired if the due date has passed.

1. **Chore notification**

Users can notify chore assignees via email.

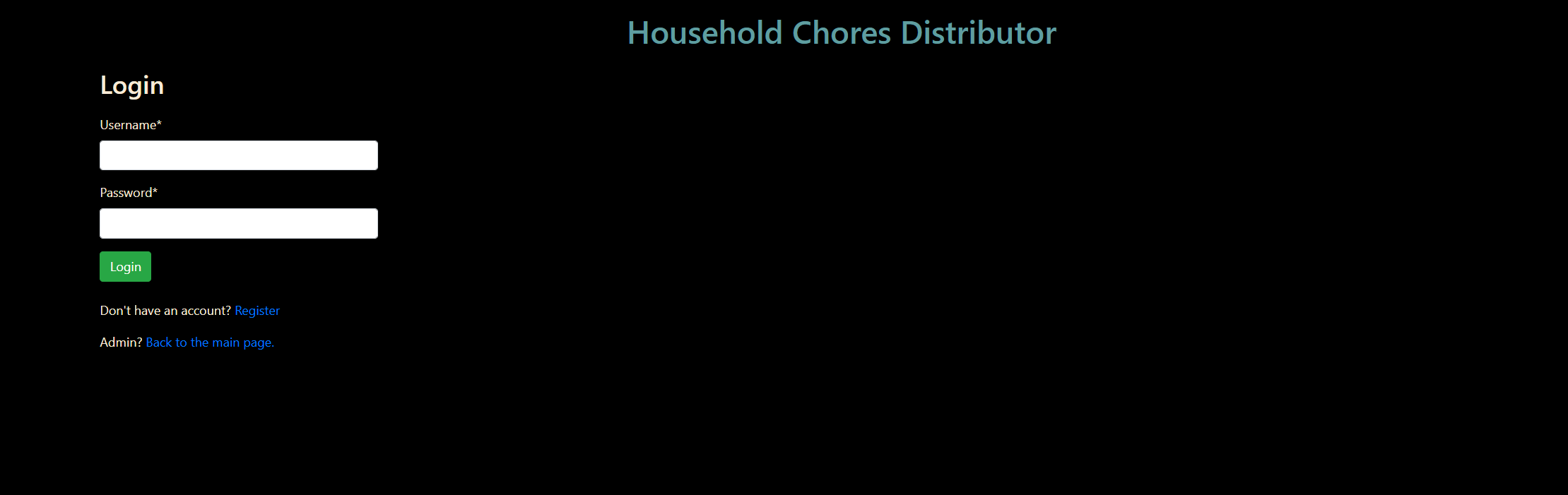
1. **Signup**

This page is used for account registration for the application.



1. **Login**

To access the application, users must first authenticate their log in credentials by providing a valid username and password.



## 1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Chore | A specific household task or responsibility. |
| Create Chore | The process of assigning a chore to a household member |
| Chore Priority | The level of importance to a chore |
| Chore Notification | Alerts and reminders for chore assignments. |
| Chore Completion | Marking a chore as finished when completed |
| Chore Delete | Option to delete a Chore |
| Customization | User-configurable settings for chore preferences and priorities. |
| User | Individuals who interact with the Household Chores Distributor system |
| User autonomy | Users will have independence and control of managing Household Chores system without need of having admin privileges. |
| Application Design Document | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. |

**Section 8 – References**

# IEEE Guide to Software Requirements Specifications - <https://ieeexplore.ieee.org/document/278253>

# Lucid Chart - <https://www.lucidchart.com/pages>