

# Automating management of shared activities in coliving spaces using Colyf

Hrishikesh Waikar, Bhaskar Gurram, Tosha Kamath, Atul Gural  
Department of Computer Engineering, San Jose State University  
hrishikesh.waikar@sjsu.edu  
bhaskar.gurram@sjsu.edu  
tosha.kamath@sjsu.edu  
atul.gural@sjsu.edu

**Abstract**— Sharing accomodation and staying together is a common way of life for many people across the globe. As students or professionals sharing an apartment, there's no dedicated person to do all the household tasks and duties. Hence, such a setup requires that the responsibilities and work is shared. And that all members do the work equally, while ensuring that all activities are completed. We have come up with a solution that helps manage these activities. Colyf provides an online tool to help create, delegate, monitor and close tasks to a set of people in a house. It also provides a tool for ordering groceries together.

**Keywords**-- Task allocation, management, coliving, equal distribution, assignment.

## INTRODUCTION

THIS document lays out the activities involved while living together in shared accomodations and how use of a management tool in the form of a software can help simplify and streamline the execution of these activities. A lot of people - students, bachelors, working professionals live in shared accomodations, especially in cities. The primary reason for doing so is to split the rent. Also they are not staying with their families. Hence such an arrangement has become quite the norm for many people. While living together, there are many household activities such as errands, cooking, cleaning activities, payment of bills, shopping, etc. that are done together. Resources are split, so are tasks. People in a house share these tasks by taking turns, or fixing certain tasks to certain people. The current systems are very arbitrary and inefficient. It requires group meetings, assigning tasks verbally with lack of accountability. In most cases, it results in one person doing more work, while some others doing it less. These things can cause issues amongst the co-tenants and result in bad experience if some people feel that tasks are not either carried out by some people or not delegated equitably.

The report assumes a particular setup of the house in which multiple people such as students are staying together, sharing rooms, common areas such as drawing rooms. They cook together for each other, order groceries or other stuff shared with each other and so on.

The activites that this tool specifically attempts to address

are of 2 types. One is all tasks that need to be done on a regular basis, ie. after a fixed duration of time such as day, week or month. Eg. Cooking or cleaning the room. Second is ordering/buying things together by making a list for it. Best example for that is ordering groceries every week. Colyf is an online web application that provides an interface for management of these activities.

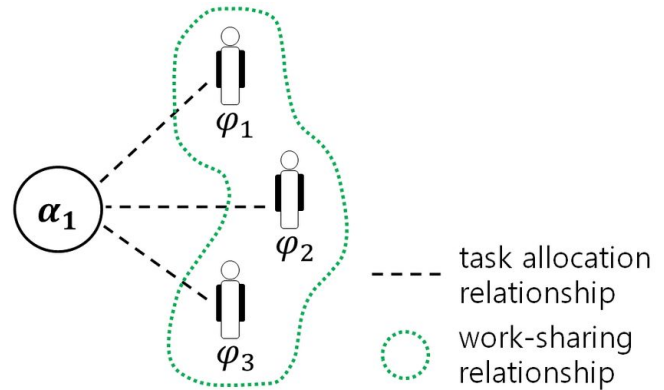
## I. TASKS

*A. The problem with task delegation the way it is being done so far*

A house has many tasks that need to be done on a regular basis that are common amongst the tenants. The common ones are cooking, cleaning of different areas of the house, paying bills, etc. There are 2 features about an activity that it must have to be addressed by our system.

1. It needs to be executed regularly after definite intervals such as days, week or month.
2. It is something that affects multiple people, hence a need to take turns to do it.

For example, a room shared by 3 students needs to be cleaned every day. Hence, taking turns makes sense. Another example is a group of 4 people cooking for each other where a person cooks for him or herself along with 3 others. The next day another person does so for all of them. All these activities involve multiple people taking turns to do a work that affects the entire group. And these activities need to be done for the length of their stay in that accomodation.

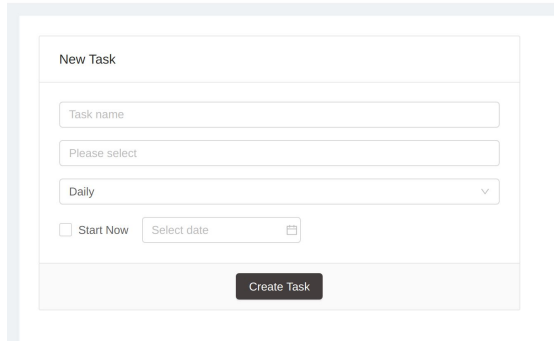


### B. The platform

Colyf is an online platform where houses can create their account. A person can create a house at signup with names and emails of the members. A verification email is sent to all the members to accept and join in. Once inside, the house members can access the task management tool.

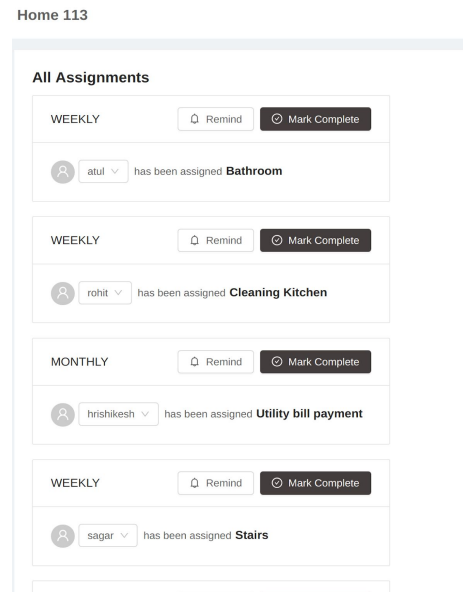
### C. Task Creation

A user can create a task by specifying a name, choosing the people from the house that are involved in it and specifying the rotation type of the task - one time, day, week or month.



### D. Task Assignment

The task once created will be allocated to a different person at the beginning of each rotation period. One can choose to do the first allocation at the time of creation of the task as well. One can view all the assignments under the all tasks tab. There are separate views to view and manage, daily, weekly and monthly tasks. A task can be reassigned to another person manually.



### E. Task Completion

A task can be marked as completed. Only Once completed, the system updates the task completion count against the person it was allocated to..

## II. THE TASK ASSIGNMENT ALGORITHM

When a task is created, it is meant to be done by the people that were selected at the time of its creation in a turn by turn basis. For example, a task to vacuum the living room carpet once every week is created with 5 members of an apartment. The task allocation must happen to one member per week, with each week having a new member.

The system maintains against each individual a task count, which is the number of times that task has been done by that person. So a person has different task counts for different tasks he/she is involved in. When the new rotation period begins (eg. for weekly tasks, Monday morning 12:00 am), each task must find a suitable candidate to assign the task to. This it does by referring to 2 metrics - task count and rotation type count. The rotation type count could be weekly count for weekly type task, daily count for daily type task.

The rotation type holds the number of tasks assigned to the person for the current duration of rotation type. So a weekly count of person 1 will hold number of weekly tasks the person has been assigned.

The task allocation algorithm attempts to find the right person to allocate a task to by referring first to the task count followed by rotation type count. For a given task, it seeks the the person who has done the task the least number of times.

If there are more than 1 person with the same lowest task count, it goes to refer to the rotation type count. Among the people with equal lowest task count, it seeks the person with lowest rotation type count. Simply meaning, which one of them has been assigned lowest number of task for that rotation period. If it finds one person, task is assigned to that person. If it finds multiple people again with same lowest weekly count, it will randomly allocate the task to one of them.

The task count is updated against a person when he/she completes the task. The rotation type counts are reset to 0 with the onset of a new rotation period.

In this way the algorithm knows which people have done the task the least number of times while also ensuring that in a given week (or any rotation period) more weekly tasks are not allocated to the same person.

As a person completes her task, her count increases and her chance to do the task again decreases as the algorithm seeks people with the lowest count.

Another aspect is how equilibrium is still maintained if a task is reassigned. When another person is assigned, when she completes the task, her task count

increases. This ensures, that person who was supposed to do the task originally has still un-increased, lower task count thus ensuring that he/she comes up again for being assigned. On the other hand the person who does the task on her behalf will be least preferred by the system during next turn.

### III. Co-ordering

Colyf also provides a feature for creating a list of items to be ordered among multiple people such as groceries. A list can be created and added with items by different people at different time before it is used for the final placement of order. The order list could include items shared by all the members, some members or completely individual items. Each list item can have a customized set of individuals from the group sharing it. This will be helpful in attribution and accounting on tools such as splitwise.

The screenshot shows a web application window titled "Add List" with a close button (X). Inside, there's a text input field containing "May 15th Groceries". Below this, there's a list of items with their quantities and the names of people who added them. Each item row has a trash icon to remove it. The items are: Bread (quantity 2, people: Bhaskar, sagar, Vinit, atul, rohit), Chicken (quantity 1, people: Vinit, atul, Bhaskar), Milk (quantity 1, people: Bhaskar, Vinit, sagar, rohit, atul, hrishikesh), Ramen (quantity 2, people: rohit, Vinit, sagar), and Broccoli (quantity 0, no people listed). At the bottom, there's an "add people" input field, an "Add More" button, and "Cancel" and "OK" buttons.

the time of writing this report. The authors would like to acknowledge the house members for being subject to experimentation of the use of the application while it affects how they are allocated tasks regularly

### REPOSITORY

The link to the github repository is:  
<https://github.com/bhaskarvgurram/CoLyf>  
<https://github.com/bhaskarvgurram/CoLyf>

### REFERENCES

- [1] <https://nodejs.org/en/>
- [2] <https://aws.amazon.com/>
- [3] <https://reactjs.org/>
- [4] <https://www.mongodb.com/cloud/atlas>

### THE SYSTEM AND PLATFORM

Colyf is deployed as a web application with the server developed in Express JS [1], a node platform deployed on AWS[2].

The client application is built in React[3]. The database is a NoSQL database of MongoDB. The Mongoose ORM layer is used for object oriented modelling and usage of the database. [4]

### ACKNOWLEDGMENT

The application is being currently used for task management at the house currently shared by the authors at