Home Management System
Project Proposal

Description
Home management portal to track home related activity such as expenses, grocery purchases, who is cooking next, whose turn is in cleaning trash using a combination of functions like expense manager, to do list, note taking utilities along with a instant messenger for member communication regarding the activity.

Users
Each project(name for new profile) consists of groups. Group can be created by one or more individual members. The portal consists of three abstract activity.

Activity
Task, List and Expense. Task involves work which will consume time of the activity based on the type of work, example Trash cleaning or cooking. List is a way to specify items in a order or unordered form, grocery listing for next week or things to repair in house. Expense activity will involve the money component in it, grocery purchases, refueling vechile. One or more combination can be used to create groups. The groups will encompass members. The task alone group will be cooking turns. Expense activity will involve trips. List activity will involve. Below is the example of each.

Scope
Each users in the given project will have equal access. The groups can be created by members. Members can create new groups, add, delete or modify items.

Tables
Login(username,password)
Members(member_id,username,member_name,email,phone)
Groups(group_id,group_name)
Member_groups(member_group_id,group_id,member_id,member_value)
Expenses(transaction_id,group_id,item,cost,expenses_shared_with)
Notes(note_id,group_id,list,status,pin_to)
Tasks(task_id,group_id,task,status)
Messages(message_id,to_id,from_id,message,time)
Schedules(schedule_id,schedule_activity,start_date,schedule_interval,iterations)

Table description

Login — Contains data for user credential verification.

Members — Will hold all the member details.

Groups — Members form a group which involve in certain activity. Expenses, Lists and Tasks uses this group as a reference to specific member activity.

Member_groups — This table reduces the redundancy of using both *group_id* and *member_id* in each activities(Expenses, Lists, Tasks) and as well as messages instead *member_group_id* is used in it's place. *member_value* column is assigned unique value to a member within a group which enables to identify if only few member in that group involved in the group or all. For example, If a purchase is made and it should be shared between 3 members in a 5 member group. *member_value* column will help in identifying these three members out of 5 and expenses will be shared among only these 3.member_value column contains value in powers of 2 to identify which member in the group is involved. For example, group

containing 5 members. Consider a,b,c,d,e are 5 members relatively assigned as a as 1st member, b as second member $\cdots$.

$$member\_value = \sum_{i=1}^{n} 2^{(i-1)} * x_i$$

$$x_i = \begin{cases} 1 & \text{if } i^{\text{th}} \text{ is involved in activity} \\ 0 & \text{if } i^{\text{th}} \text{ is not involved in activity} \end{cases}$$

For the example,

$$\begin{aligned} \textit{only member a is involve} &= 2^0 * 1 + 2^1 * 0 + 2^2 * 0 + 2^3 * 0 + 2^3 * 0 = 1 \\ \textit{only member a and e is involve} &= 2^0 * 1 + 2^1 * 0 + 2^2 * 0 + 2^3 * 0 + 2^3 * 1 = 1 + 16 = 17 \\ \textit{only member a,c and e is involve} &= 2^0 * 1 + 2^1 * 0 + 2^2 * 1 + 2^3 * 0 + 2^3 * 1 = 1 + 4 + 16 = 21 \\ \textit{all members are involved} &= 2^0 * 1 + 2^1 * 1 + 2^2 * 1 + 2^3 * 1 + 2^3 * 1 = 1 + 2 + 4 + 8 + 16 = 31 \end{aligned}$$

This table needs both SQL and NoSQL database implementation to accommodate the Message table.

Expenses Activity involving expenses. *expenses_shared_with column* references *Member_groups(member_value)* to identify whether the expenses is of individual or whole group or few members in a group.

Notes,Tasks Activity involving Lists and Tasks. *Shared_with column* refers to who is a group this tasks is assigned to. Absence of pin_to refers to tasks completed by that member.

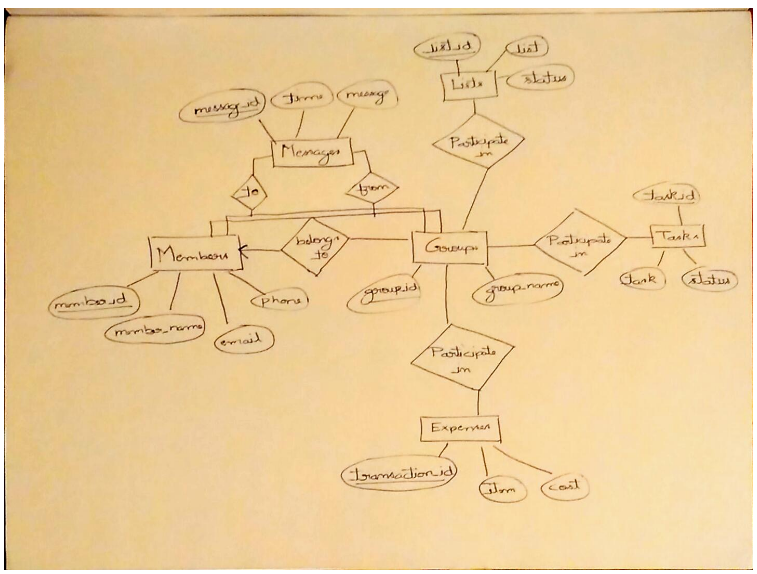
Messages Message can be attached to any activity, group or individual based on *to_id*(sender of message by an individual) and *from_id*(reciever of the message to group or individual or selected members in a group for an activity). The given relation is given for a SQL database which will be translated to NoSQL database upon implementation.

Schedules Attaching the scheduling event to the activity(Expense,Notes,Task) or to a Group. Scheduling can be one time or recurring at a constant interval. *interval* signifies the time gap between two activity. *iterations* specifies number of times this activity should be repeated.

Entity Relationship Diagram

Implementation
Database implementation is using PostgreSQL and MongoDB(Messaging and member_groups table alone). User interface is either using html/php or J2EE(JSP) or AngularJS.

Dataset
For the project the personal available data is used having 7 years of personal expenses, trips, grocery bills. The available data is modified to suite to the project. Remaining data will be created.



list_id
list
Lists
status
message_id
time
message
Messages
Participate in
to
from
Participate in
task_id
Tasks
Members
belongs to
Groups
task
status
member_id
member_name
phone
email
group_id
group_name
Participate in
Expenses
transaction_id
item
cost