Project Part 2 - GROUP

Team: Deepak Kyasaralli Thimmappa
Rashmi Shetty
Yawen Zhang

Title: Life Wisely

Project Summary:

A web application to manage life wisely. The application provides reminders on things which are often neglected, including diet, medication, hygiene, stress buster etc. Besides the reminder function, a log system is in place to keep track of the achievements and a reward system is in place to encourage completion of tasks.

Project Requirements:

Business Requirements			
ID	**Requirements**	**Topic Area**	**Priority**
BR-01	Each login is via CU email	Authentication	Critical
BR-02	Tracking the activity of users in certain period and producing logs	DB	High
BR-03	Giving rewards for user's achievements in certain period	Reward	High
BR-04	Software upgrade (should only happen on weekends)	Upgrade	Good-to-have

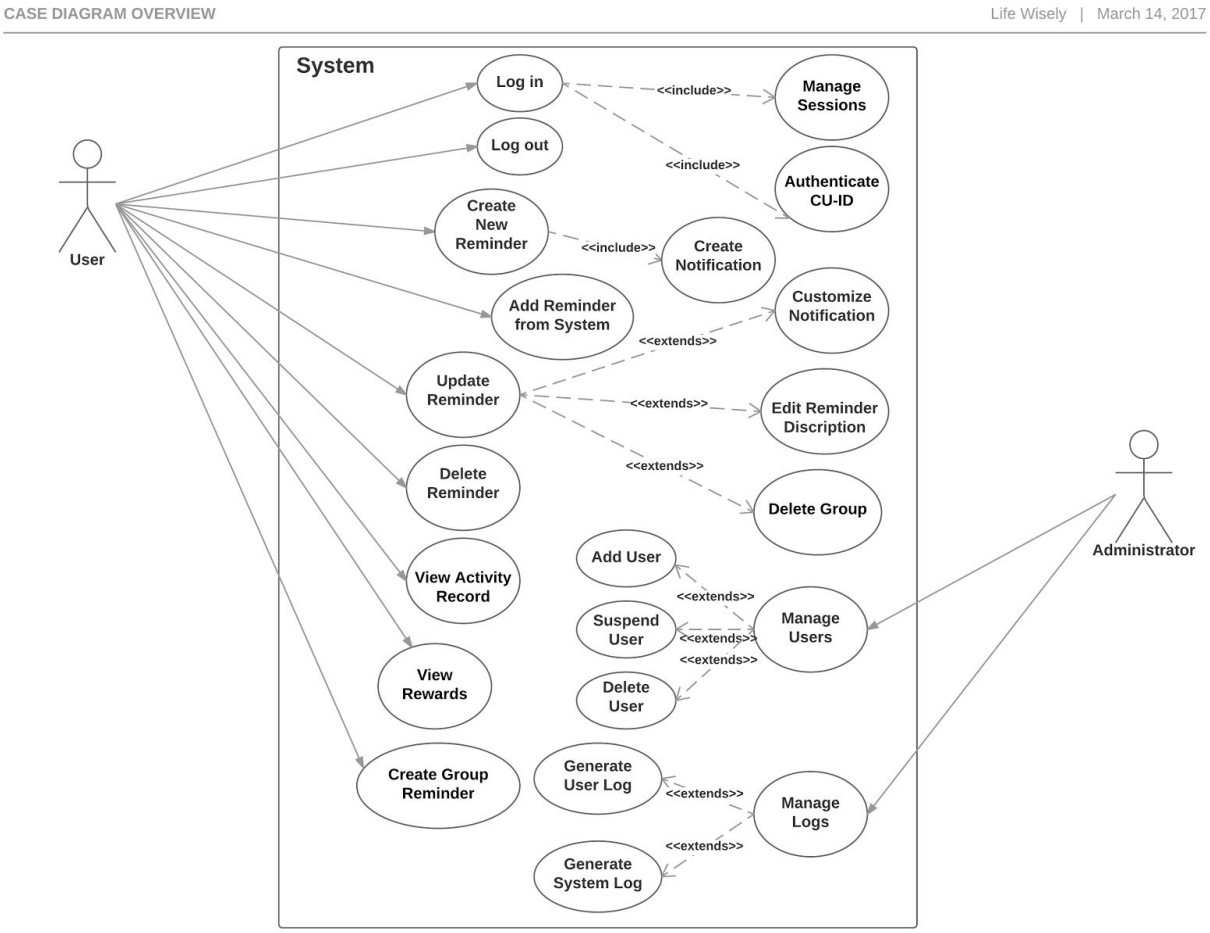
User Requirements				
ID	**Requirements**	**Topic Area**	**Actor**	**Priority**
UR-01	As a user, I want to login the system	Authentication	User	Critical

UR-02	As a user, i want to logout the system	Authentication	User	Critical
UR-03	As a user, I want to create a new reminder	Reminder	User	Critical
UR-04	As a user, I want to customize reminder notification	Reminder	User	Critical
UR-05	As a user, I want to add existing reminder from system	Reminder	User	Critical
UR-06	As a user, I want to delete a reminder	Reminder	User	Critical
UR-07	As a user, I want to update a reminder	Reminder	User	Critical
UR-08	As a user, I want to create a group reminder	Reminder	User	High
UR-09	As a user, I want to delete group related to a reminder	Reminder	User	High
UR-10	As a user, I want to view my regular activity record	DB	User	High
UR-11	As a user, I want to view rewards for my achievements	Reward	User	High

UR-12	As a system administrator, I want to add user	DB	Admin	High
UR-13	As a system administrator, I want to delete user	DB	Admin	High
UR-14	As a system administrator, I want to suspend user	DB	Admin	High
UR-15	As a system administrator, I want to generate user log	DB	Admin	High
UR-16	As a system administrator, I want to generate system log	DB	Admin	High

Non-Functional Requirements			
ID	**Requirements**	**Topic Area**	**Priority**
NR-01	300 concurrent users should be supported	Performance	High
NR-02	Access by all OS	Platform	Critical
NR-03	Access to external data (import from Google calendar)	Interface	High
NR-04	Backup DB	Security	High

Use Case Overview:



Data Storage:

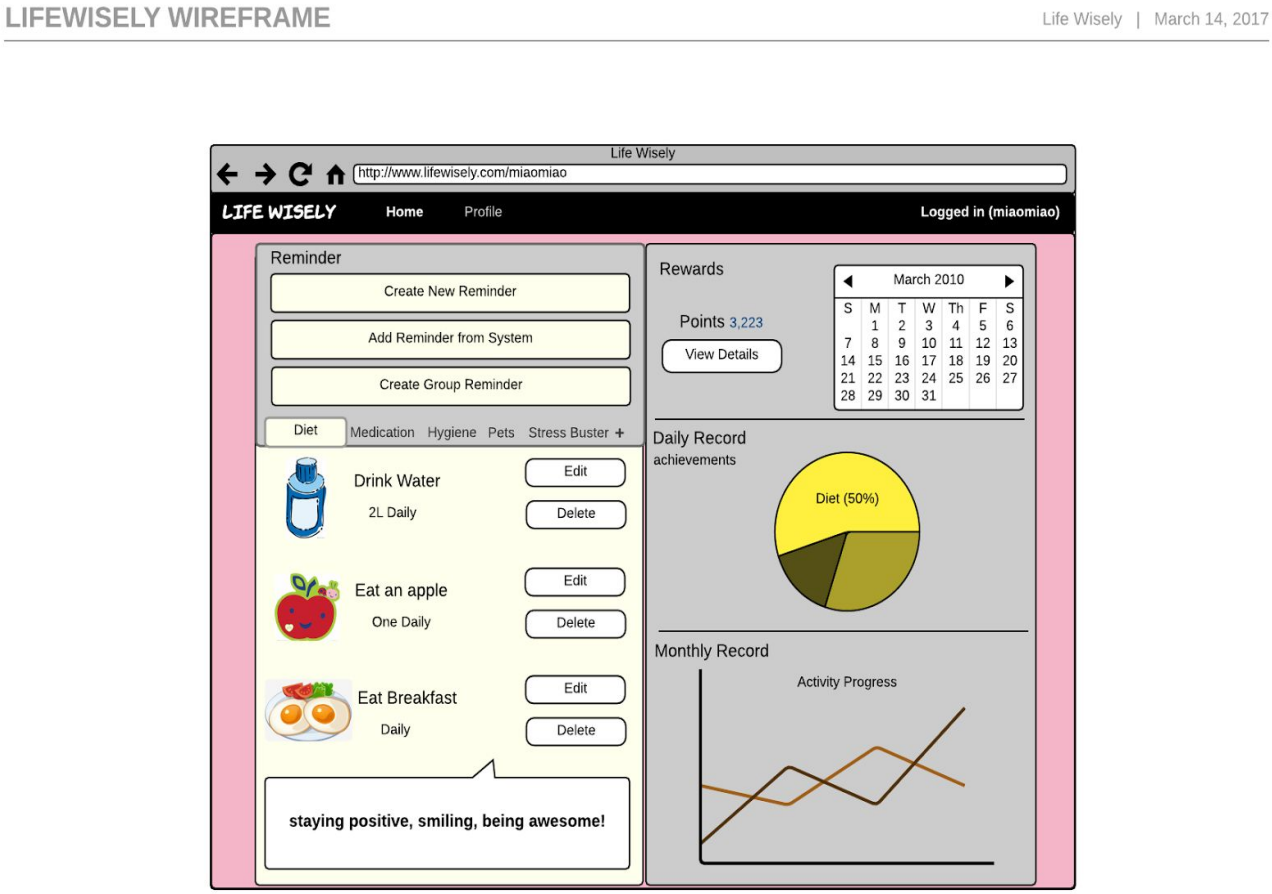
- DB: PostgreSQL
- Data is distributed in 6 tables

User information is stored in User table with columns as User email,Password,User Name, Gender, Phone number, Email having User email as primary key.
User group information is stored in User Group table with columns as Admin User (user who creates the group), member User Email, Group name and simple description.
Reminder information is stored in Task table with columns as User email,, TaskID, Task Name, Begin Time, End time and status with User email as foreign key

Log information table which stores User email, log in time and log out time with User email as foreign key.
Notification table with Task ID, Notification title and Recurrence type with Task ID as foreign key.
Reward table with Task ID, points with Task ID as foreign key.

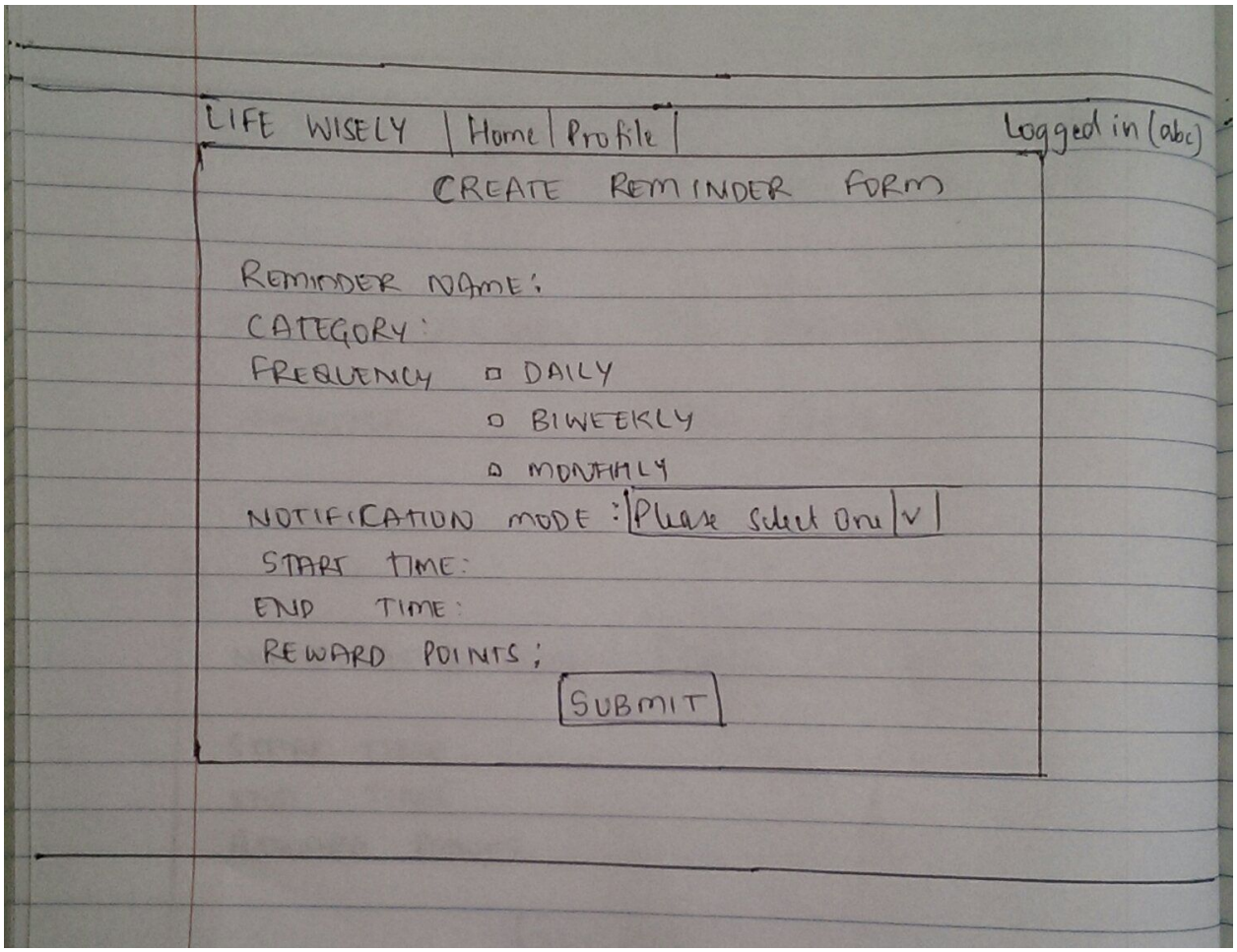
UI Mockups:

- **Main UI (after user logged in)**

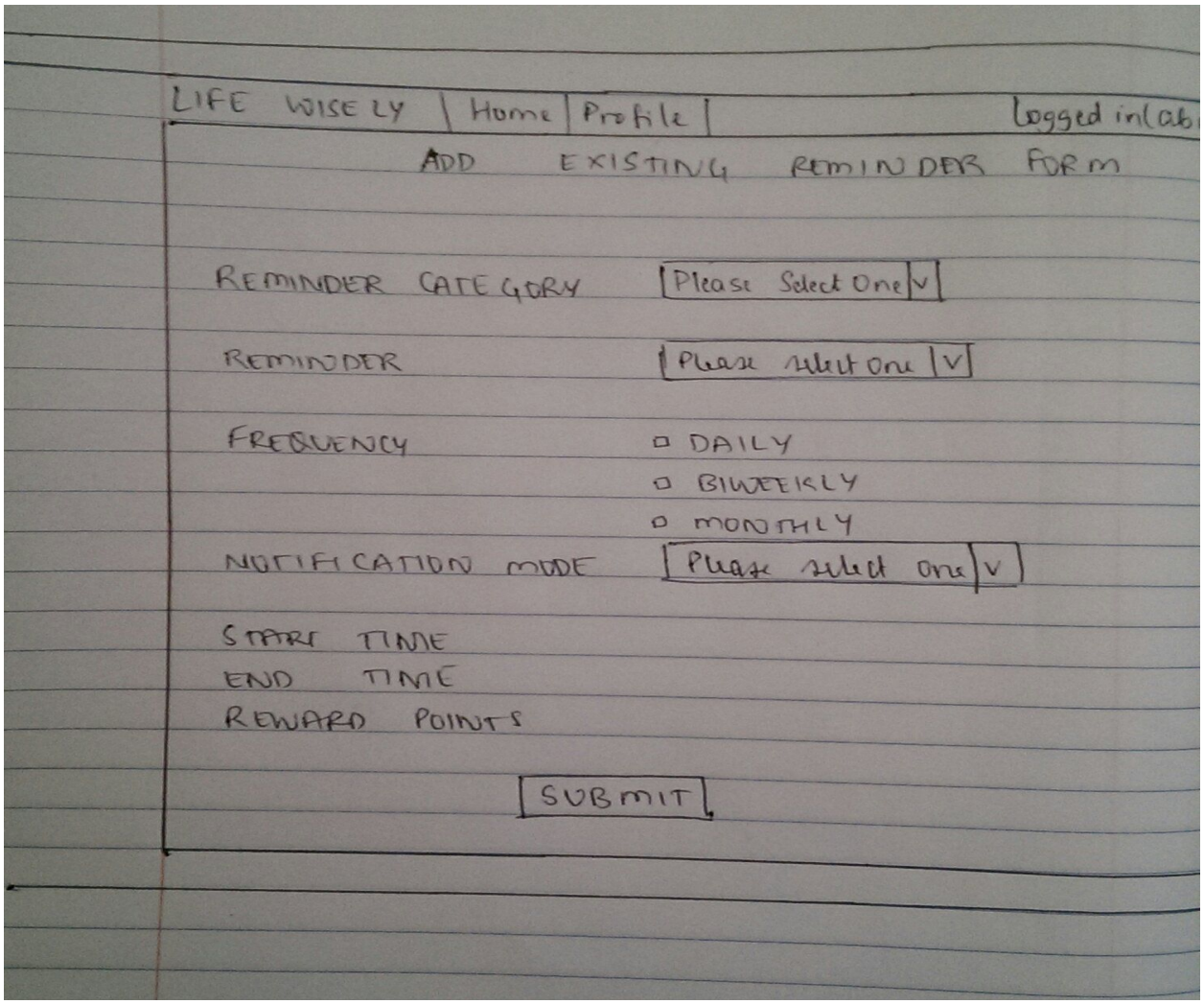


- **UI flow**

i)When user clicks, Create New Reminder, the following form is displayed

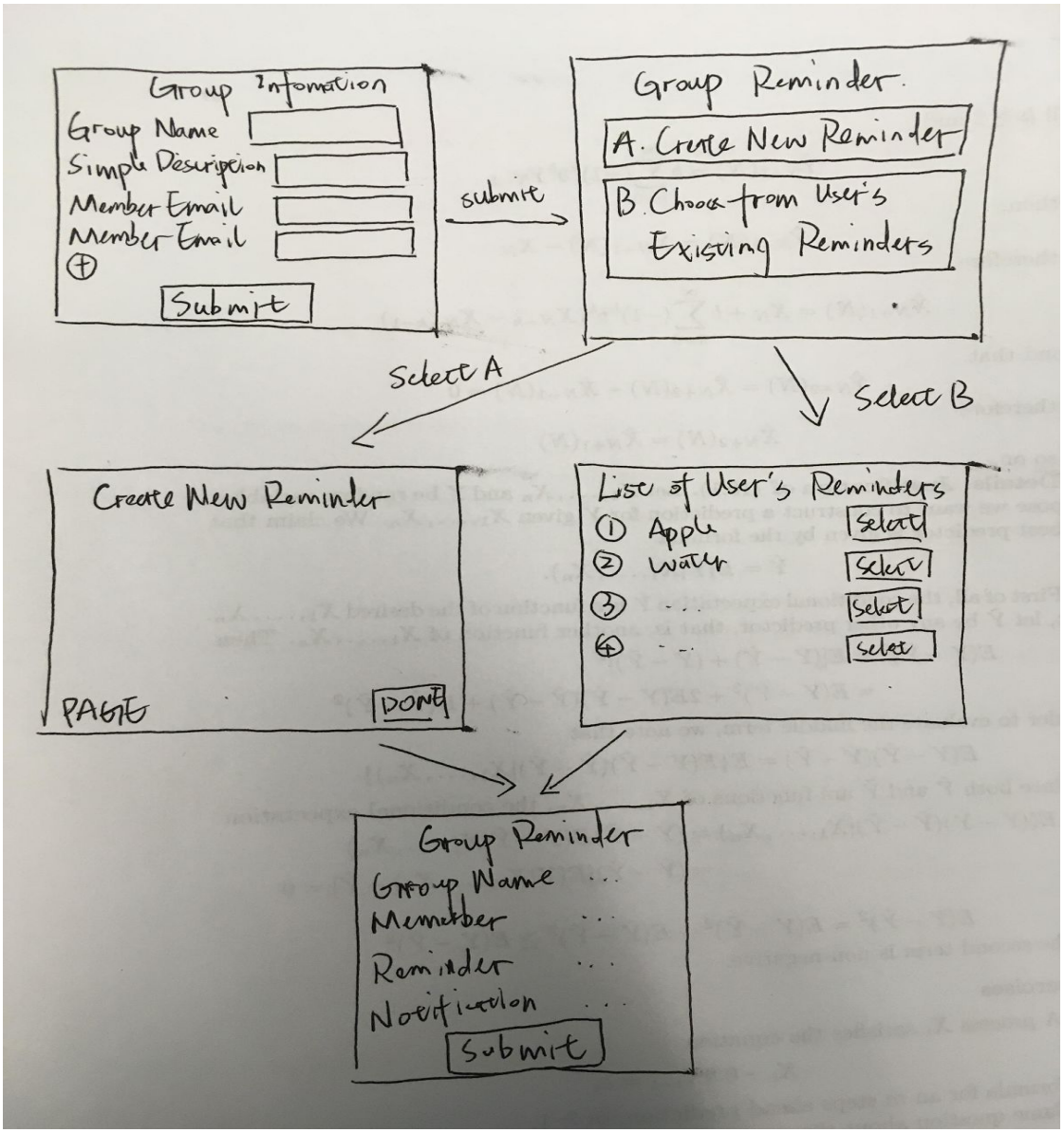


Submission is confirmed with home page loaded with updated task
ii)When user clicks, add existing reminder form, following form is displayed:

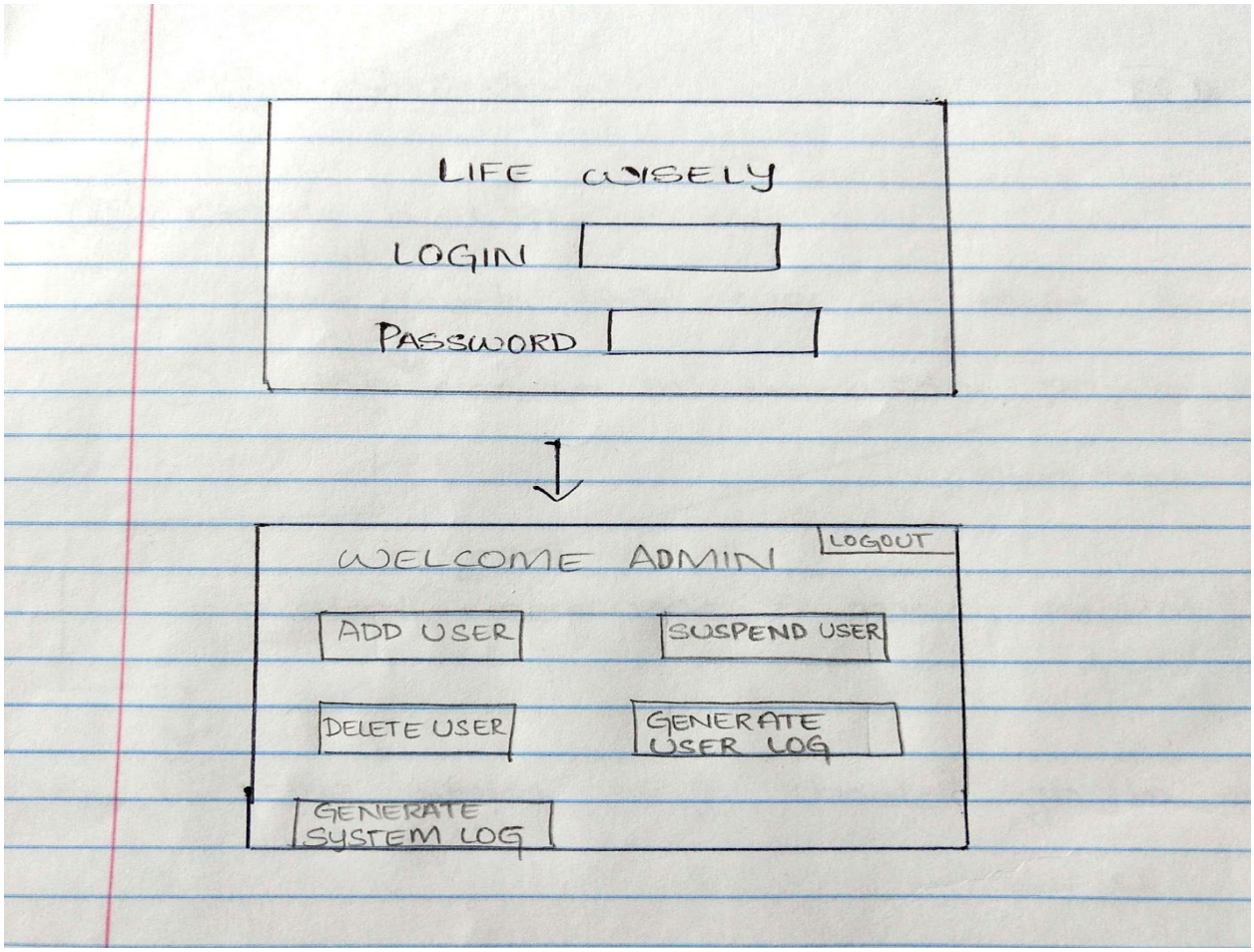


Upon completion of either forms, the user is presented with updated home page.

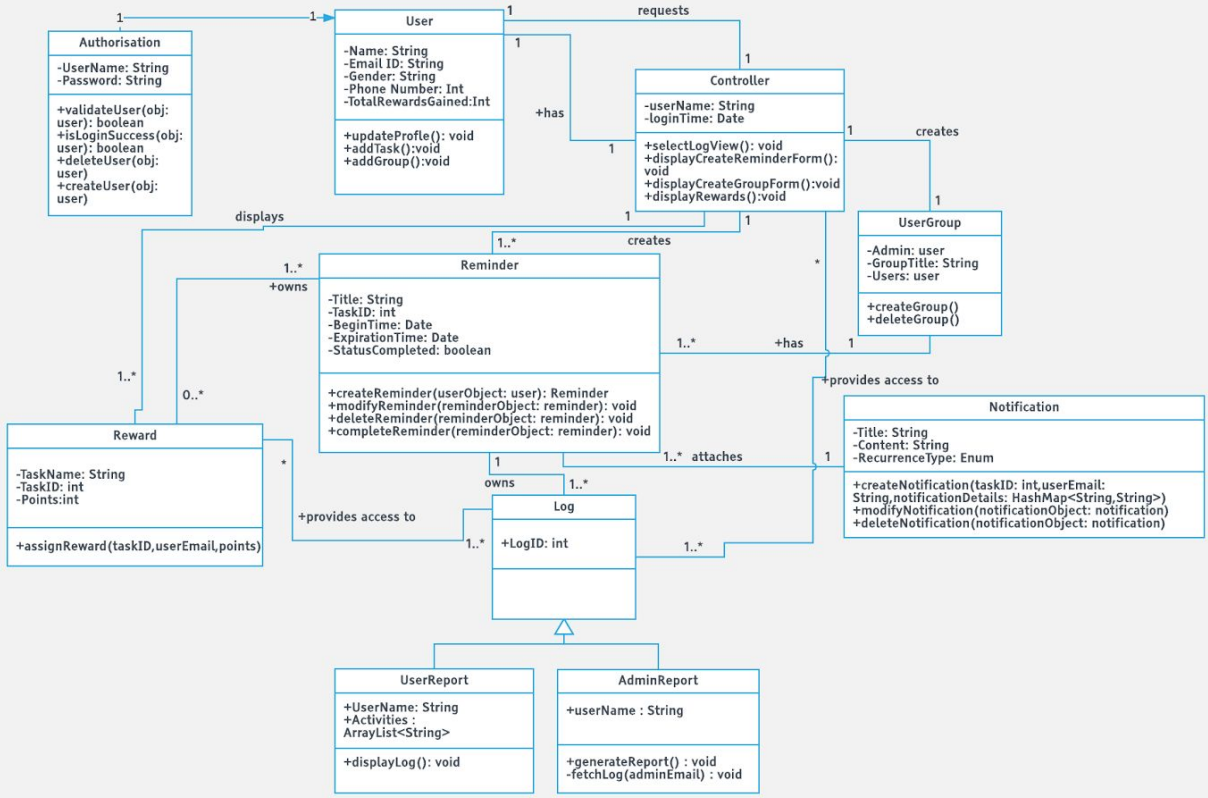
iii)When user clicks, create group reminder, the following page is presented. Upon completion of initial form, the user is provided option of creating reminder or select existing reminder as mentioned in i) or ii) of UI-flow


Group Information
Group Name
Simple Description
Member Email
Member Email
Submit
submit
Group Reminder
A. Create New Reminder
B. Choose from User's Existing Reminders
Select A
Select B
Create New Reminder
PAGE
DONE
List of User's Reminders
1 Apple
2 Water
3 ...
4 ...
Select
Select
Select
Select
Group Reminder
Group Name ...
Member ...
Reminder ...
Notification ...
Submit

Admin View:



Class Diagram:


Authorisation
-UserName: String
-Password: String
+validateUser(obj: user): boolean
+isLoginSuccess(obj: user): boolean
+deleteUser(obj: user)
+createUser(obj: user)
User
-Name: String
-Email ID: String
-Gender: String
-Phone Number: Int
-TotalRewardsGained:Int
+updateProfle(): void
+addTask():void
+addGroup():void
requests
+has
Controller
-userName: String
-loginTime: Date
+selectLogView(): void
+displayCreateReminderForm(): void
+displayCreateGroupForm():void
+displayRewards():void
creates
UserGroup
-Admin: user
-GroupTitle: String
-Users: user
+createGroup()
+deleteGroup()
displays
creates
Reminder
-Title: String
-TaskID: int
-BeginTime: Date
-ExpirationTime: Date
-StatusCompleted: boolean
+createReminder(userObject: user): Reminder
+modifyReminder(reminderObject: reminder): void
+deleteReminder(reminderObject: reminder): void
+completeReminder(reminderObject: reminder): void
+owns
+has
+provides access to
Notification
-Title: String
-Content: String
-RecurrenceType: Enum
+createNotification(taskID: int,userEmail: String,notificationDetails: HashMap<String,String>)
+modifyNotification(notificationObject: notification)
+deleteNotification(notificationObject: notification)
attaches
Reward
-TaskName: String
-TaskID: int
-Points:int
+assignReward(taskID,userEmail,points)
+provides access to
owns
Log
+LogID: int
UserReport
+UserName: String
+Activities : ArrayList<String>
+displayLog(): void
AdminReport
+userName : String
+generateReport() : void
-fetchLog(adminEmail) : void