CSCI E-64 Mobile and Cloud Computing

Final Project Functional Specifications

Luke Walsh

**Overview**

For my final project I propose to build a social, household- or group-based mobile app primarily for roommates, but also potentially for just groups of friends, which allows for exchanges of a virtual currency. The currency, which I’ll call “karma points” for now, is distributed based on a voting system within the household, in which users may request points or propose a vote to impose a cost or a “fine” on other household members. The main use case is for a group of roommates who may be interested in keeping track of who contributed to the household (ex. doing chores) or caused stress for the household (ex. paying their share of the bills late), with the points as a rough representation of whether a member is in good standing with the group or needs to contribute more. The point of the app is 50% functional and 50% for fun (i.e. not to be taken completely seriously).

Users can create a household or join a household by being voted in unanimously by the rest of the group. A user may belong to multiple households and has a karma balance for each household he or she belongs to, and the goal of the app is to stay in a positive karma balance by doing good things for each household, and avoid negative karma balances and the associated shame by not doing things that affect others negatively (such as not doing the dishes).

All users start at a karma balance of 0, and the net balance for a household is always 0. Household members will go into positive karma when they do something good and vice versa. When a member believes he or she has done something good for the household he or she can request an amount of karma to be “paid” to them, and submit the request to the household at large for a majority vote. The request then can be viewed on the other members’ phones, and they can each vote whether they want to approve or deny the request. If a majority of the members vote to approve it, the requestor’s balance will go up by that amount and the amount will be “paid” for by deducting the amount from the other members’ accounts evenly (so for the very first request for the household, for example, the other members will go into a negative balance).

Conversely, we’ve all had that roommate who can never remember to take the trash out when it’s their turn (or help shovel the driveway, or remove their hair from the drain, etc…). One feature the app will have is the ability to propose a “fine” on a member in the group. The proposal is then viewable by other household members’ mobile devices, and if it’s approved by a majority the offender will be deducted that amount which will go into the other members’ accounts evenly. The app will also support a messaging forum so roommates can communicate with each other.

The app will have a home screen where a user must first sign up/login in, and then be able to easily view which households they belong to, create a new household, or view households to which their Facebook friends belong, and clicking on a household will bring up that household landing page. There will be several screens for each household. If the user is not part of the household, there will be a button to request to join the household (the request will show up in that household’s voting queue and needs to be approved by vote by the members). If the user is a member of the household they will have access to the following screens: a household main screen which contains basic info about the household and the ability to vote for a Landlord or evict a member of the household, a screen to view the requests that are currently being voted on (i.e. active proposals but have not yet reached majority threshold), and one for a household message board where users can view and post messages (optionally anonymously) to the household at large.

These screens and functionalities are outlined below.

**Home Screen**

This is the screen that will display on app launch. The purpose of the home screen is for users to easily view which groups they belong to, create a new household, and view households their Facebook friends belong to.

* If the user is not yet logged in yet, the home screen will show the user a button to sign in or register using Facebook. The app will use Facebook authentication.
* Once the user is logged in, the following will appear:
  + A scrollable list of households the user belongs to. Clicking on one of these households will bring the user to that household’s main screen.
  + A scrollable list of households the user’s Facebook friends using the app belong to (but is not yet a part of). Clicking on one of these households will bring the user to that household’s main screen (the user won’t have full access yet- see household screen’s section). This is how users can join existing household (they must be Facebook friends first).
  + A button that gives the user an option to create their own household. When a user creates a household, he/she automatically becomes the “Landlord” role for that household and has the following abilities that the regular household members do not have (these can be used in the household-specific screens, described in the next section):
    - Change the name and description of the household.
    - Change the name of the “karma” currency being used.
    - Change the background picture for the household’s main screen
    - Veto (i.e. cancel) any vote before it passes the majority threshold, except for votes related to voting for a new Landlord.
    - The Landlord may be voted out if a proposal to elect a new Landlord gets a majority vote (the new person will become landlord).

**Household Main Screen**

The purpose of the household main screen is to provide a basic overview of the household, the user’s current karma balance, and a list of the current household members.

* Display the following info about the household:
  + Household name (editable by Landlord)
  + Household description (editable by Landlord)
  + Name of currency (editable by Landlord)
  + Name of the Landlord
  + Full, scrollable list of members of the household with the following info:
    - Name of the member (from Facebook)
    - Current karma balance
    - “Vote to elect Landlord” button. When clicked, this button creates a new proposal to make the member the landlord and will then appear in the proposal list for the household. It will appear greyed out and unclickable if the member is the current landlord or if there is already a proposal in the list to make the member the landlord. A unanimous vote among the remaining members (i.e. all members except the current Landlord) is needed in order to have a successful “mutiny”
    - “Vote to evict” button – creates a new proposal to evict the member from the household and adds the proposal to the list of proposals for the household (viewable on the voting screen). A unanimous vote is needed for this type of proposal (excluding the member being evicted).
* Two navigation buttons at the bottom, or perhaps tabs:
  + View votes – brings the user to the voting screen
  + View message board – brings the user to the group message board

**Household Voting Screen**

The purpose of this screen is to allow the user to view all the requests/voting proposals currently being voted on (i.e. they’ve been proposed but haven’t reached the majority threshold for passing or failing), and propose a new karma vote/request.

* Display a scrollable list of all the current proposals/votes that are in progress as well as votes that have passed or failed. Each proposal/request will have the following information:
  + Type of proposal (i.e. “Karma”, “New Landlord”, “New Member Approval”, “Evict Member”)
  + Status of the vote (ex. “2 members for, 1 against, 3 needed for majority” – 3 needed for a household of 5 for example for a karma vote)
  + If a karma proposal, display the member’s name to which the proposal applies and amount of karma to be added/deducted
  + If a new Landlord proposal, display the proposed Landlord’s name
  + If a new member proposal, display the member’s name
  + Brief one-line message/description
  + Name of the member who proposed it (or display “anonymous”)
  + Upvote and downvote buttons. Once a member votes it will be greyed out and become unclickable for him or her.
* A top section of the screen will give the user the ability to propose a change in karma for someone (including themselves). These requests can be in a positive or negative amount and can optionally be done anonymously:
  + Dropdown or scroll menu of current members to select a member to which the proposal will apply
  + An input box for the amount of karma (positive amount means the member will be granted additional karma that will be added to that member’s account; a negative amount represents a “fine” that will be subtracted, if approved by majority members. The added/deducted amount will come from the other members’ accounts, maintaining a 0 net household balance)
  + A message line for a brief description about the request (ex. “for cleaning the dishes”, or “b/c Matt forgot to turn out the lights before leaving again”). There will be some sort of character limit.
  + Checkbox for submitting request anonymously
  + A submit request button
  + A cancel button
  + Once the user submits the request it will appear in the above list of current proposals
* Two navigation buttons at the bottom:
  + View Household main screen – brings the user to the voting screen
  + View message board – brings the user to the group message board

**Household Message/Alerts Screen**

The purpose of this screen is to act as a common place for users to create and view messages that are viewable to the entire household. The messages are also searchable.

* A main window that contains a list of member-generated messages posted to the thread for the entire group to see. These messages will have the following info:
  + Message text
  + Name of the sender
  + Timestamp
* A text window underneath the message window where users can type in a message for the rest of the household to see.
* A Submit Message button
* Two navigation buttons at the top:
  + View Household main screen – brings the user to the voting screen
  + View votes – brings the user to the voting screen

**All screens:**

All screens will have as part of the navigation bar a button to go to the home screen.

Below is a brief description on how the project will meet the requirements:

# Final Project Core Requirements

Your final project **must** incorporate the following:

1. Azure Mobile App Services and/or ASP.NET Web API using Azure API Apps
   1. Be sure to describe why you choose one or the other or both The app will use a mobile app services backend
2. Azure SQL The app will use Azure SQL to store all user info, karma “account” balances, household info, voting info, etc.
3. REST Interfaces
   1. Must be used for all Azure hosted services you create
   2. Must not be state-full such that you require the caller to contact the same compute node
   3. The app will use REST interfaces for all services
4. Authentication
   1. Shall be supported using one or more of the following
      1. Microsoft Account
      2. Facebook
      3. Twitter
      4. Google
      5. Windows Azure Active Directory
   2. The app will use Facebook authentication
5. Authorization
   1. You must have roles that control
      1. Access to certain functions and REST interfaces
      2. Visibility of specific UI elements
   2. You must have at least two roles
      1. Admin Role  
         Access for administrative functionality  
         (Approving self-service user enrollment for example)
      2. User Role  
         Standard user functionality  
         (Adding a stock to monitor for example)
   3. There will be two roles for each household- a standard member role which allows for all normal voting functions, and a “Landlord” role which allows additional abilities for that user:
      1. Change the name and description of the household.
      2. Change the name of the “karma” currency being used.
      3. The Landlord may be voted out if a proposal to elect a new Landlord gets a majority vote (the new person will become landlord).
6. Azure Web Jobs – All voting functions will be supported with web jobs as well as uploading/processing the household background picture (not implemented)
7. Azure Blob or Azure File Storage The app will use Azure Blob to store the background picture for households. (not implemented)
8. Secure Services
   1. SQL Injection attacks
   2. Un-authenticated access to costly resources
   3. We haven’t learned about this yet but it will be incorporated
9. Azure Mobile Client Application(s) for one of the following platforms
   1. Universal App using C#
      1. Can be a Desktop, Phone, Tablet or Raspberry Pi 2 IoT core app
   2. Android App using Xamarin and C#
   3. iOS App using Xamarin and C#
   4. The app will be designed for Android using Xamarin and C#
10. Unit tests for ALL
    1. REST Services you create
    2. The app will be unit tested (not implemented)

# Electives

Your final project **must** also include **three** of the following elective technologies:

1. Azure Tables
2. Azure Queues
3. Azure Redis Cache
4. Azure Document DB The app will use Azure Document DB to store the user messages
5. Azure Notification Hubs
6. Azure Worker Roles
7. Azure Event Hubs
8. Azure IoT Hub
9. Azure API Management
10. Azure Search – the app will use Azure Search to allow users to search through the household messages
11. Azure Websites