**Azure Classes and Controllers:**

* Startup
  + Sets up routing configuration for all controllers
  + Initializes database context using Entity Framework Code First
  + Adds several test users, households, and household members to the context
* Friend Controller
  + Primary role is to retrieve the households of Facebook friends of the current user who also use the app, but the user is not yet a member of. This is accomplished via an authenticated GET request.
  + The friend controller is called on the app’s main/welcome screen after the user logs in to allow the user to see the households of his or her friends that also use the app. The user needs to view these households in order to request to join them.
  + If the user is already a part of a friend’s household, this household will not appear in this request’s results.
* Household Controller
  + The primary role of the household controller is to get, create and update data related to households. A household is the basic communal space that users can be a part of. The relationship is a user can have many household memberships (HouseholdMember), and a household can have many memberships. Two other functionalities are also associated with households- voting and messages. Household members can participate in voting and messaging which is associated with a particular household. All household members can view and participate in the voting and messaging.
  + GET request that returns a list of households that a user is a member of.
  + GET request that returns a specific household by its id with the following info:
    - Household name
    - Household description
    - Name of currency
    - Name of the Landlord
    - List of household members
  + GET request that returns a household membership (HouseholdMember) based on a household id and user id.
  + POST action that creates a new household. This is called from the client app's home screen when the user would like to create a new household. The user that created the household will automatically become the landlord. Returns a message indicating success or failure of the creation.
  + POST action that updates the basic info about a household such as the household's name, a description, and the name of the currency. This method is called from a household's home screen to allow the landlord to edit these fields. Only the landlord may edit these fields:
    - Household name
    - Household description
    - Name of currency
  + GET request that returns list of household members for a specified household id.
* Message Controller
  + The primary role of the message controller is to store and retrieve messages for each household in Azure Document DB. Household members can create/post and view messages for a household, which are viewable to the entire household. Document DB is useful for this purpose because it allows for flexible schema which is useful if other data fields should eventually be added to the messaging functionality, and it also allows for Azure Search which will allow users to search prior messages.
  + GET request that returns all of the messages in the collection
  + GET request that returns all messages for a specified household.
  + POST action that creates a new message in Azure Document DB
* Registration Controller
  + The primary role of the registration controller is to register a user, i.e. save their info in Azure SQL, once they've authenticated through Facebook. This allows their Facebook user ID and name to be stored in Azure SQL rather than having to make a call out to Facebook's Graph every time that info is needed. The user does not have to initiate this action: once they authenticate through Facebook these methods are automatically called by the client app.
  + GET method that returns the registration status of the current user.
  + POST method that allows any authenticated user to register. This method is called automatically by the client app after the user logs in / authenticates with Facebook. If the user already exists no action is taken, but if it’s a new Facebook user their info is added to the Azure SQL database.
* User Controller
  + Get list of all current users (for dev/testing only) via get request
  + Get any user by user id via get request
  + Return the current user
  + Return a list of households of the user’s Facebook Friends who are also using the app via a Get request, but that the user is not a part of
* Notification controller:
  + Return notification preferences for the user for a given household via a get request, and allow a user to update these preferences via a post request:
    - Receiving push notifications for updates on requests/votes (vote increase, request approved, request failed, etc).
    - Checkbox for receiving push notifications for new messages posted by members to the message board.
    - Checkbox for receiving push notifications for changes to the household info (name or description change, members joining or leaving).
* Voting controller
  + Return list of all votes for a given household via a get request:
    - 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)
    - Member the vote applies to (member if karma vote or landlord vote, user if a new member vote)
    - Member who proposed the vote (or anonymous)
    - Brief one-line message/description
    - List of member ids that voted for/against the vote
  + Allow user to propose a vote for the household via a post request. If the user is not a member of the household, return a bad request:
    - Household ID
    - Type of proposal (i.e. “Karma”, “New Landlord”, or “Evict Member”)
    - Member to which it applies
    - If karma proposal, amount of karma to be added/deducted
    - Whether proposal should be made anonymously
    - Brief one-line message/description
  + Allow users of a particular household to cast a vote for an existing vote. Each member is only given one vote and may not take back their vote. The controller will accept a vote containing:
    - The id of the vote
    - The vote, “For” or “Against”