Initial list of functional specs

**Non-Member Expectations**

1. **Create An Account**
   1. The website should allow a non-member user to create an account. The expected inputs are username, password, and ZIP code.  
      The website will not allow the user to be created if the username already exists within the database. There will also be a password re-entry field. The website will prevent user creation if both password fields do not match. The website will provide error feedback if the ZIP code is not a valid ZIP.
   2. **Stimulus/Response Sequence – Account Creation**
      1. User enters desired UserID
      2. User enters password
      3. User re-enters password for validation
      4. User enters their default ZIP code
      5. Website will check that UserID is unique, passwords match, and ZIP is valid
      6. Website will store usename, encrypted password, and ZIP in the database layer
      7. Website will inform user that account was created
      8. Website will redirect to homepage as a logged in user
   3. **Functional Requirement Label**
      1. REQ 1.1 Create An Account

**Member Expectations**

1. **Use Forums** 
   1. Registered users can use the forums to view and post new comments and threads. The system will not allow unregistered users to view the forums. Registered and authenticated users can use the forums.
   2. **Stimulus/Response Sequence - View Forums**
      1. User navigates to forums
      2. Website will display various subforums and threads
   3. **Stimulus/Response Sequence – Search Forums**
      1. User can search for posts with keywords – this may be out of scope for the final product.
   4. **Stimulus/Response Sequence – Create New Post/Reply**
      1. User can either reply to an existing thread or create a new one
   5. **Functional Requirement Label**
      1. REQ 2.2 View Forums
      2. REQ 2.3 Search Forums
      3. REQ 2.4 Post to Forums
2. **Play Games**
   1. Registered users can play games. Unregistered users cannot access the games page. Users can choose from various games. User progress/scores can be tracked in the database layer if development time allows.
   2. **Stimulus/Response Sequence – Play Games**
      1. User navigates to games tab
      2. User can choose from games to play
   3. **Functional Requirement Label**
      1. REQ 3.1 Play Games
3. **Use Image Sharing** 
   1. Registered users can view and post images to share with other registered users. Comment may be made to the images found here.
   2. **Stimulus/Response Sequence - View Images**
      1. User navigates to image sharing tab
      2. Website will display previously posted images
   3. **Stimulus/Response Sequence – Post Images**
      1. User navigates to image sharing tab
      2. User can upload their own image
   4. **Stimulus/Response Sequence – Comment on Images**
      1. User navigates to image sharing tab
      2. User selects an image to comment on
      3. User posts a comment
   5. **Functional Requirement Label**
      1. REQ 4.2 View Images
      2. REQ 4.3 Post Images
      3. REQ 4.4 Comment on Image
4. **Display Local Data – Weather**
   1. The website will display at least one widget containing weather information for the current user. Data will be sourced using the registered users ZIP code, if valid. If not, widget will default to Boca Raton weather forecast.
   2. **Stimulus/Response Sequence – Display Weather**
      1. Widget will display weather if a user is logged in.
      2. No input required. Widget will be based on the users ZIP code.
   3. **Functional requirement Label**
      1. REQ 5.2 Display Weather
5. **Access COVID Resources**
   1. Registered users can access various COVID related resources.
   2. **Stimulus/Response Sequence – View Resources**
      1. User selects Resources tab. Website will populate with various external COVID resources.
   3. **Functional requirement Label**
      1. REQ 6.2 Resources

Non-Functional Specs

**Compatibility Requirements:**

1. *Browsers:* The site will be created using bootstrap to ease compatibility between desktop and mobile based browsers. The initial scope of the site covers compatibility with Chrome and Firefox, which should also provide compatibility with other major browsers. No alternative site functionality will be considered if the browser does not have JavaScript installed.
2. *Computer and OS:* The site is being built with no particular operating system in mind. If the OS can run any popular browser, it will run the site.

**Expected Load:**

1. As the site should not be receiving large use traffic simultaneously, we will make all possible accommodations to tolerate multiple users at once. Scripts will be executed to simulate multiple simultaneous users/requests to the server.

**Storage Requirements:**

1. System will be stored on LAMP server. Backup options are not yet determined but may include cloud storage for image and database files.

**Availability Requirements:**

1. System will be stored on LAMP server. As such, it is subject only to LAMP server maintenance constraints.
2. In the event of LAMP downtime, website will be inaccessible.

**Performance Requirements:**

1. *Responsiveness:* The system will be coded using bootstrap which should help accommodate any screen size.
2. *Test Requirements:*  Testing will include all functional requirements as well as simulated multiuser load and response time.
3. *Reliability:* The system should be operational 100% of the time as long as LAMP is up and running. Failures should be addressed prior to deployment.
4. *Bug Count:* No more than 10 bugs during development at any time. No more than 5 bugs when system is deployed.

**Security Requirements:**

1. *Login/Password:* Users will be required to authenticate via username and password. Passwords will be salted and hashed on the database layer to protect user privacy and security as much as possible.
2. *Test Requirements:*  Testing will include all functional requirements as well as simulated multiuser load and response time.
3. *Reliability:* The system should be operational 100% of the time as long as LAMP is up and running. Failures should be addressed prior to deployment.
4. *Bug Count:* No more than 10 bugs during development at any time. No more than 5 bugs when system is deployed.