|  |
| --- |
|  |

|  |
| --- |
| **Pixogram v4.0 UI/UX (Phase1)** |
| Case Study |
|  |
| This document covers Software Requirements of Pixogram, along with list of Technologies to be used to develop this Software System, and also includes some details on the Architecture |
|  |
| **IIHT** |
| **2/7/2019** |
|  |

Table of Contents

[1. Business Requirement(Pixogram) 3](#_Toc28724577)

[NOTE: Features marked as optional are not mandatory for GenCs 4](#_Toc28724578)

[Overview of Fields used in User Registration 4](#_Toc28724579)

[Overview of fields used for Add Content 5](#_Toc28724580)

[2. Design Inputs 6](#_Toc28724581)

[3. UI/UX Design 6](#_Toc28724582)

[Angular Components 6](#_Toc28724583)

[Components 7](#_Toc28724584)

[Sign In Components 7](#_Toc28724585)

[Sign In Component Requirement 7](#_Toc28724586)

[Sign-In Component Wireframe 7](#_Toc28724587)

[Register Component Wireframe 7](#_Toc28724588)

[Upload Media Component 8](#_Toc28724589)

[Upload Media Component Requirement 8](#_Toc28724590)

[Upload Single Media Component Wireframe 8](#_Toc28724591)

[Upload Multiple Media Component Wireframe 9](#_Toc28724592)

[My Media Component Requirement 9](#_Toc28724593)

[My Media Component Wireframe 10](#_Toc28724594)

[Media Detail Component 10](#_Toc28724595)

[Media Detail Component Requirement 10](#_Toc28724596)

[Media Detail Page Wireframe 12](#_Toc28724597)

[Followers/Following Component 12](#_Toc28724598)

[Followers/Following Component Requirement 12](#_Toc28724599)

[Followers/Following Component Wireframe 14](#_Toc28724600)

[Account Component 15](#_Toc28724601)

[Activity Log/Newsfeed Component 15](#_Toc28724602)

[Blocked Users Component 15](#_Toc28724603)

[Account Details Component 16](#_Toc28724604)

[Search Component 17](#_Toc28724605)

[4. Full Stack Technologies 18](#_Toc28724606)

[5. Technical Spec – Solution Development Environment 18](#_Toc28724607)

[5.1. Front End Layer 18](#_Toc28724608)

[5.2. Middle Tier Layer 18](#_Toc28724609)

[5.3. Database & Integration Layer 18](#_Toc28724610)

[5.4. Ancillary Layer 19](#_Toc28724611)

[5.5. Security 19](#_Toc28724612)

[5.6. Deployment & Infrastructure 19](#_Toc28724613)

[5.7. Editors 19](#_Toc28724614)

[6. Deliverables of this Phase 19](#_Toc28724615)

[7. Important Instructions 19](#_Toc28724616)

# Business Requirement(Pixogram)

Pixogram is a Social Media portal, which lets Users upload, add effects to Pictures and other Media. Users can manage the Gallery created out of uploaded Media Content. Media can be shared with the Followers. Followers can comment or like the Shared Images.

Below are the features which need to be supported by Pixogram:

The Pixogram (Single Page Picture Sharing Application) allows you to:

1. Register as a user

2. Login as a user

3. Retrieve password(when Forgot)

4. Manage your (user) account

5. Login/Logout to/from your account on Pixogram

6. Add Media Content

a. Upload single/multiple pictures, caption and description

b. Upload single/multiple videos, caption and description(optional)

7. Manage Content

a. Organize Picture in Gallery

b. Organize Videos in Playlists(optional)

c. Rename Pictures

d. Edit Caption, Description, Comment

8. Social Features

a. Use emojis in comment(optional)

b. Like or Unlike, comment, pictures and videos(optional) of other users

c. Follow/Unfollow other users

9. Hide Pictures/Videos

10. Activity/Newsfeed

a. View activity log of user-activity(i..e posted Images) on the PixoGram. That means a User’s Newsfeed shows all the Images posted by Users whom current User follows, in reverse chronological order.

11. Offline Functionality(optional):

a. Certain parts of the application should be available in absence of connectivity.

b. Relevant areas on the screen should display “Connectivity Not Available”

12. BONUS REWARDS/SCORE Feature(optional)

a. To implement offline image upload functionality such that user can upload content when offline. It will sync with backend when connected.

### NOTE: Features marked as optional are not mandatory for GenCs

### Overview of Fields used in User Registration

The application will consist of 7 fields. Given below are the fields and validation guidelines (as used in creation of UI. Some of the guidelines given for the fields in this section may not be applicable to the Java layer).

1. First Name:

a. Should allow alphabets only

2. Last Name:

a. Should allow alphabets only

3. Username

a. Should allow mix of alphabets and number

b. Username must not start with number

c. Length of username should be between 8 to 12

4. Email

a. Must allow email in valid email format

b. Must not allow two @ symbols

5. Password

a. Must be alphanumeric

b. Should allow only following special characters- . # % $ !

c. Length of password should be between 8 to 12

d. Should contain at-least one capital alphabet

6. Confirm Password

a. Should be like the above password

b. Same validation rules should apply

7. Upload profile picture

a. Upload the profile picture. Picture should be of dimension 200x200 before upload

Spreadsheet Wireframe: Empty form (Do not create in project. FYI only.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| First Name | : |  | | |
|  |  |  |  |  |
| Last Name | : |  | | |
|  |  |  |  |  |
| User Name | : |  | | |
|  |  |  |  |  |
| Email | : |  | | |
|  |  |  |  |  |
| Date Of Birth | : |  | | |
|  |  |  |  |  |
| Password | : |  | | |
|  |  |  |  |  |
| Confirm Password | : |  | | |
|  |  |  |  |  |
| Profile Picture | : |  | Browse |  |
|  |  |  |  |  |
|  | Submit |  | Reset |  |

### Overview of fields used for Add Content

There are two scenarios for content input:

1. Single Image Input

a. Title – can be alphanumeric. The length should not go beyond 80 characters.

b. Description – can be alphanumeric. The length should not go beyond 144 characters.

c. Image name – can be alphanumeric. You must supply full image name (e.g. imagesample.jpg)

d. Date – It should take current date and time using Date object.

e. The program will response with success or failure depending on whether image was saved in the database or not.

f. If success, program will end.

g. If failure, program will re-start.

2. Multiple Image Input

a. Title – can be alphanumeric. The length should not go beyond 80 characters.

b. Description – can be alphanumeric. The length should not go beyond 144 characters.

c. Image name – can be alphanumeric. You must supply multiple image names separate by comma “,” (e.g. imagesample1.jpg, imagesample2.jpg etc)

d. Date – It should take current date and time using Date object.

e. The program will response with success or failure depending on whether multiple images was saved in the database or not. Here, each image saved will have same title and description as input above.

f. If success, program will end.

g. If failure, program will re-start.

# Design Inputs

Next sections in this document provides inputs on designing the solution for above requirements.

Design inputs provided in this document are just for your reference purpose, Associates can make changes or additions to the Design, based on their analysis.

# UI/UX Design

In this Phase you will develop, responsive UI of the application using HTLM5, HTML5 API, CSS3 and Bootstrap/Material, including the Angular Components

## Angular Components

1. As per the navigation bar (each is independent page). Each page can be thought of as independent component with few child components where required:
   1. Upload Media Component
      * 1. -> Single Media Upload Component
        2. -> Multiple Media Upload Component
   2. My Media Component
      * 1. -> Media Detail Component
   3. Followers/Following Component
      * 1. -> Follower Page -> Follower Media Detail Component
        2. -> You Follow User Page -> You Follow Media Detail Component
   4. Account Details Component
      * 1. Sign In Component
           1. Blocked Accounts Component
           2. Newsfeed Component
           3. Account Update Component
           4. Search Component
           5. Logout Component
        2. Register Component

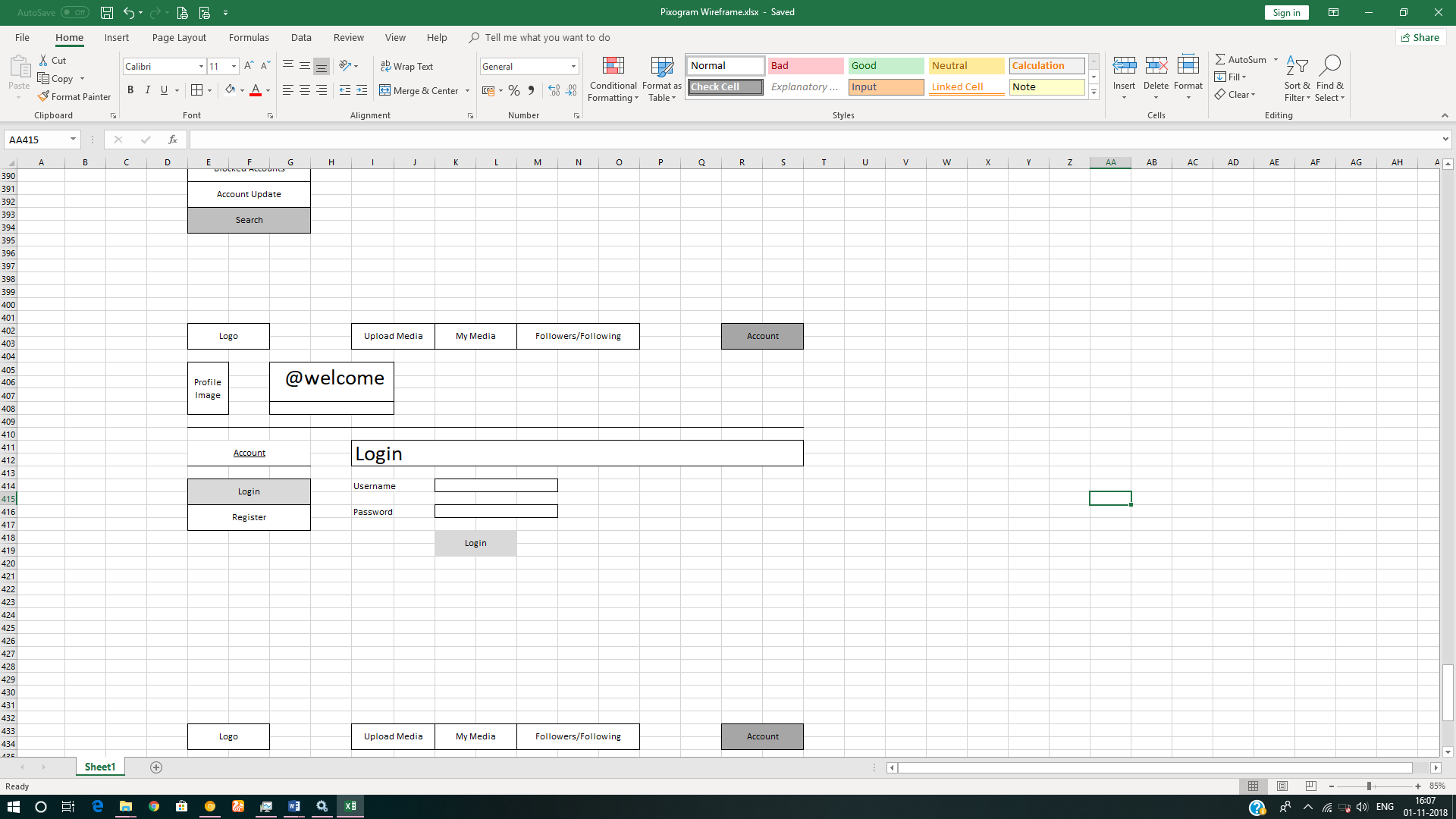
# Components

## Sign In Components

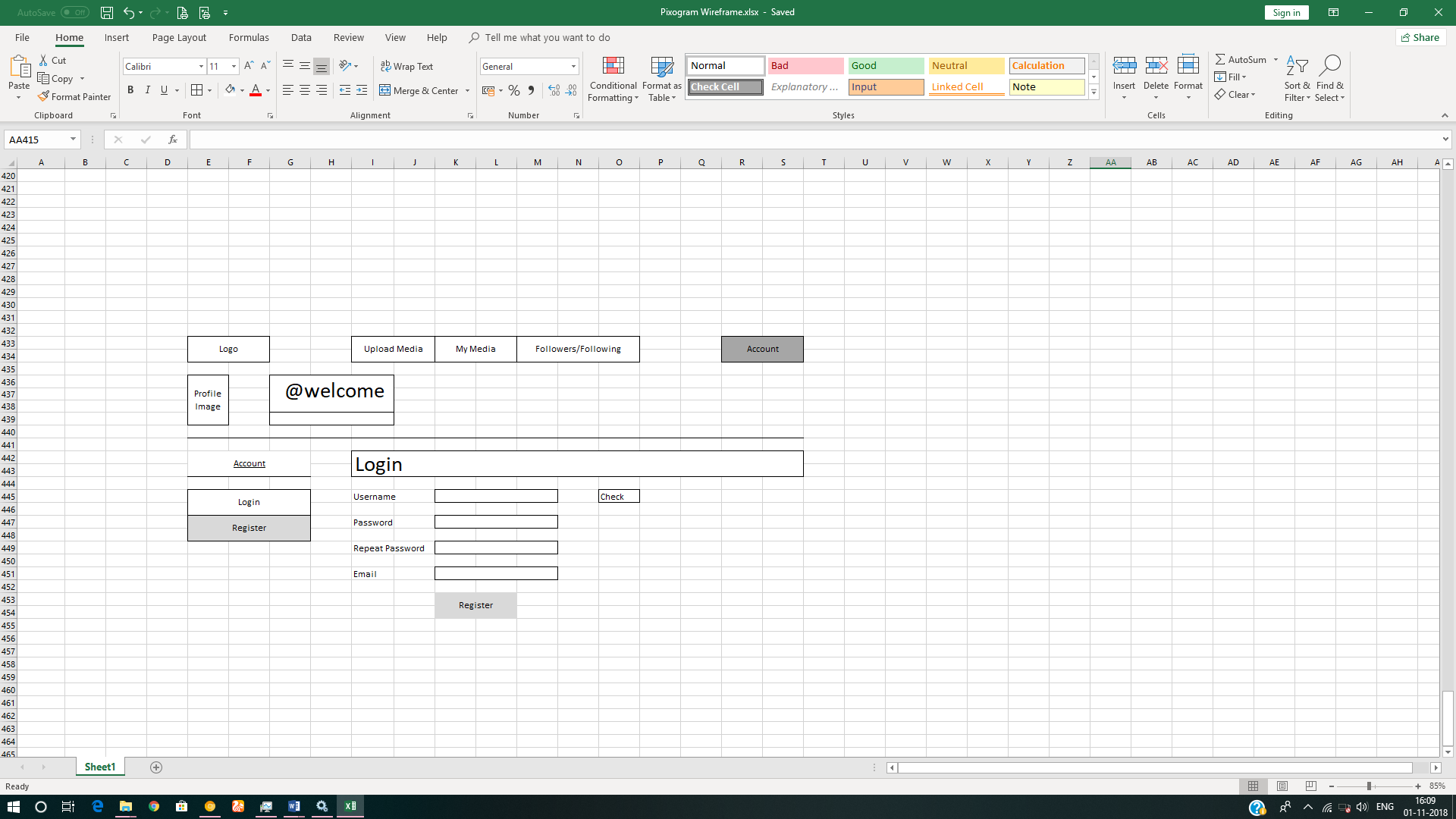
### Sign In Component Requirement

1. It allows user to sign-in with registered credentials.
2. If the user is not registered, user may register before signing-in.
   1. Username.
   2. Password
   3. Email
3. Clicking on any link: Upload Media, My Media, Followers/Following will redirect users to Login component.
4. On register component, there is check button to check if username is already in use.

### Sign-In Component Wireframe



### Register Component Wireframe

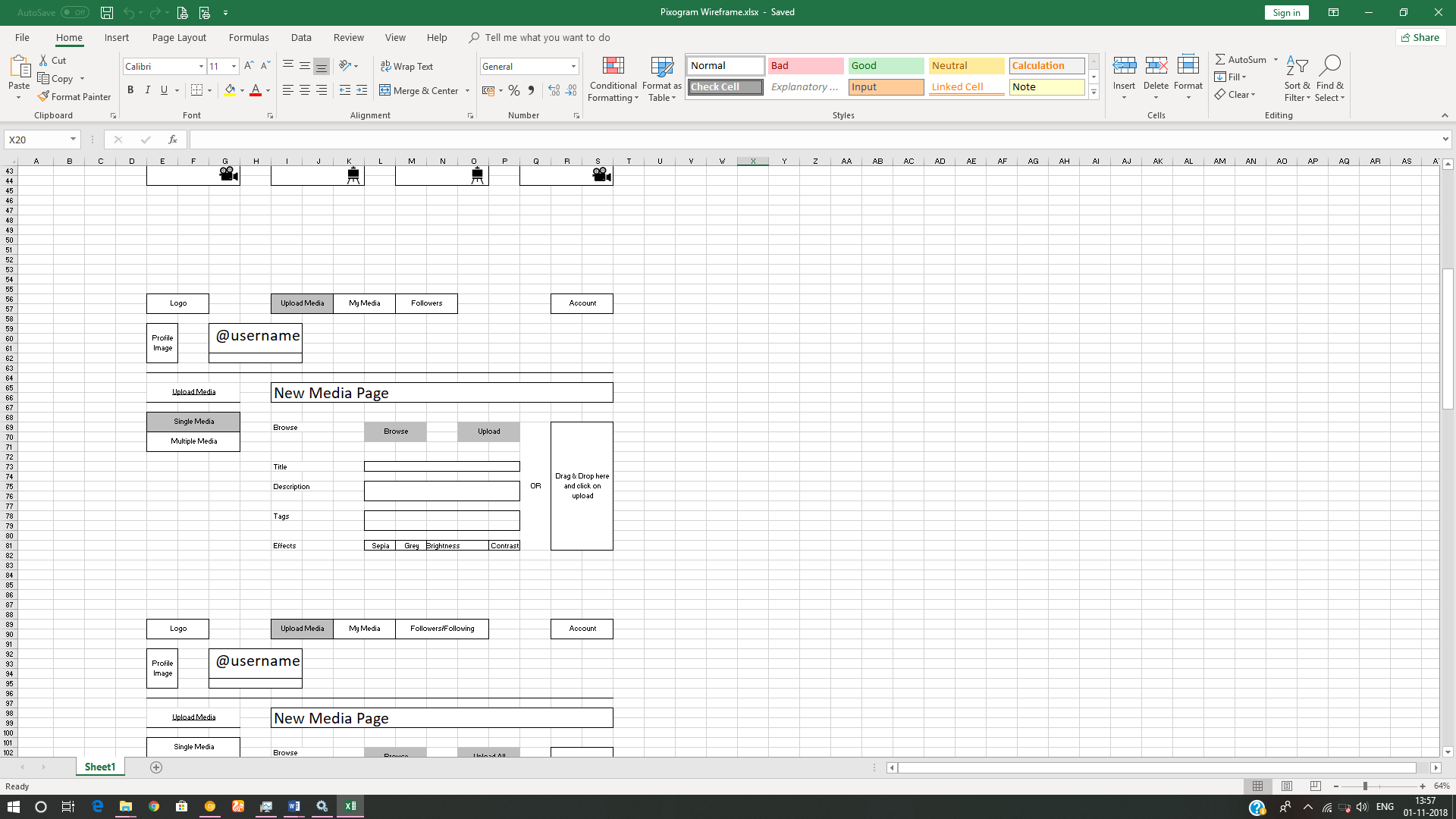


## Upload Media Component

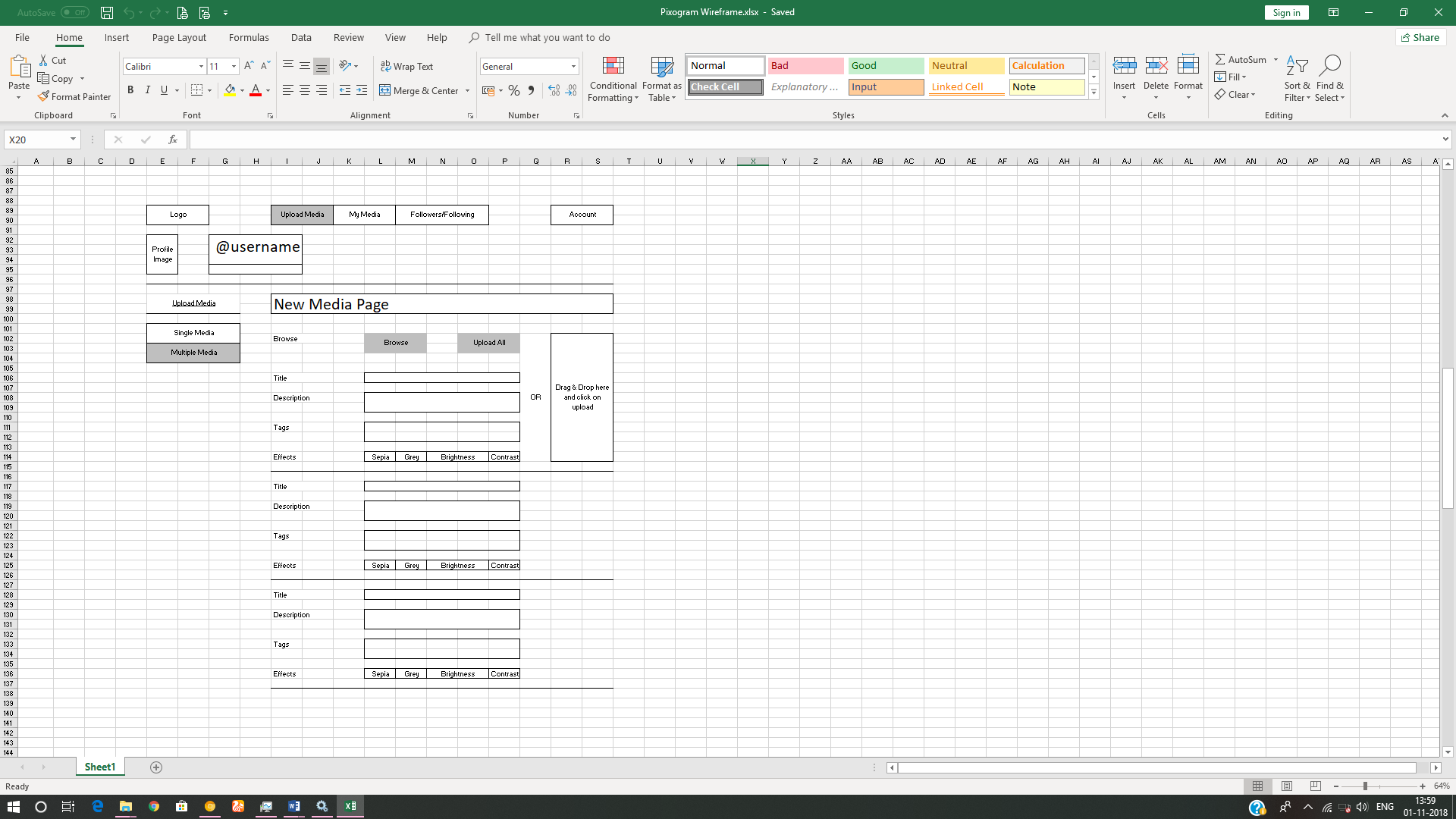
### Upload Media Component Requirement

1. It will have two sub-components
   1. Single Media Upload Component
   2. Multiple Media Upload Component
2. It allows you to upload media in two formats
   1. Images - png, jpeg, gif
   2. Video – wmv, avi, mp4
3. User should be able to upload single/multiple media items using drag and drop from file explorer in the host operating system. It is recommended that you should first create the component for single media upload. Once it is done and approved, then create the component for multiple media upload.
4. The first image which you upload will be used as a default profile picture for your account.
5. In case of video being uploaded, default image should be used as a poster/thumbnail.
6. Each upload item should have following three fields:
   1. Title
   2. Description
   3. Tags
   4. Effects – sepia, greyscale, brightness, contrast etc.
      1. Should be disabled initially. Enabled only after the media is uploaded and saved.
7. User should be able to add multiple tags; each separated by comma (,)
8. User should be able to save the uploaded media item/s
9. You will post the data to json-server using Angular httpClient library.

### Upload Single Media Component Wireframe



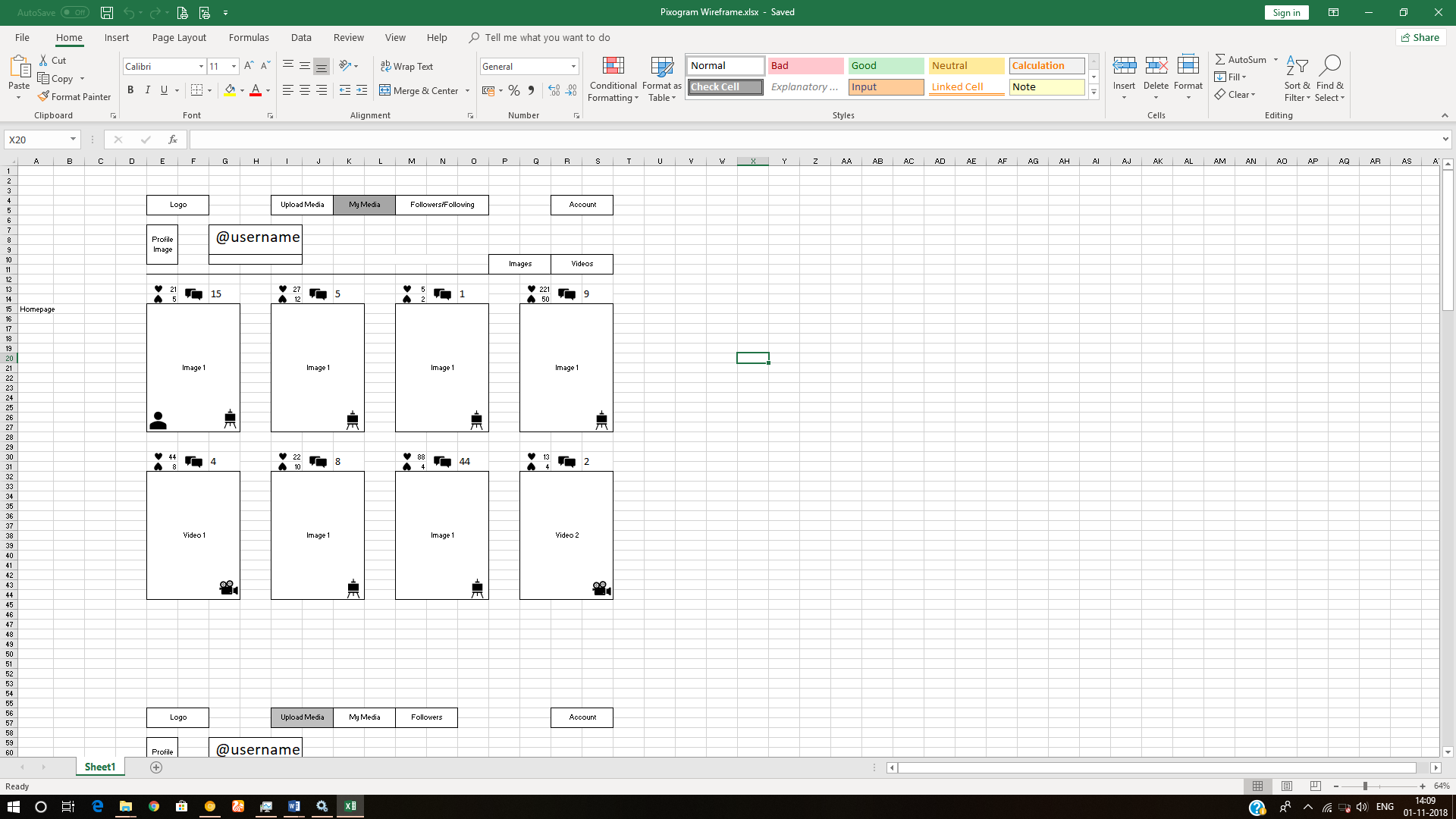
### Upload Multiple Media Component Wireframe



### My Media Component Requirement

1. This component contains all the media uploaded by you along with other information.
2. It will display your username on top along with (Follow/Unfollow) toggle button. Any user and click on Follow/Unfollow button to follow or unfollow you. It will be disabled for you as you are the account owner.
3. It will display all media items uploaded by you, as a user, in a grid format.
4. It will contain two more toggle button i.e. Images, Videos
5. If “Images” is activated, then only images are displayed.
6. If “Videos” is activated, then only videos are displayed.
7. By default, both are activated.
8. Each media item will be displayed in one cell of responsive grid with following information:
   1. Emoji Icon + number of like. (not clickable)
   2. Emoji Icon + number of unlike. (not clickable)
   3. Emoji Icon + number of comments.
   4. Emoji Icon to specify whether it is used for default profile picture.
9. User should be able to click on the media (image/video) thumbnail to view further media details and interact with the media.

### My Media Component Wireframe

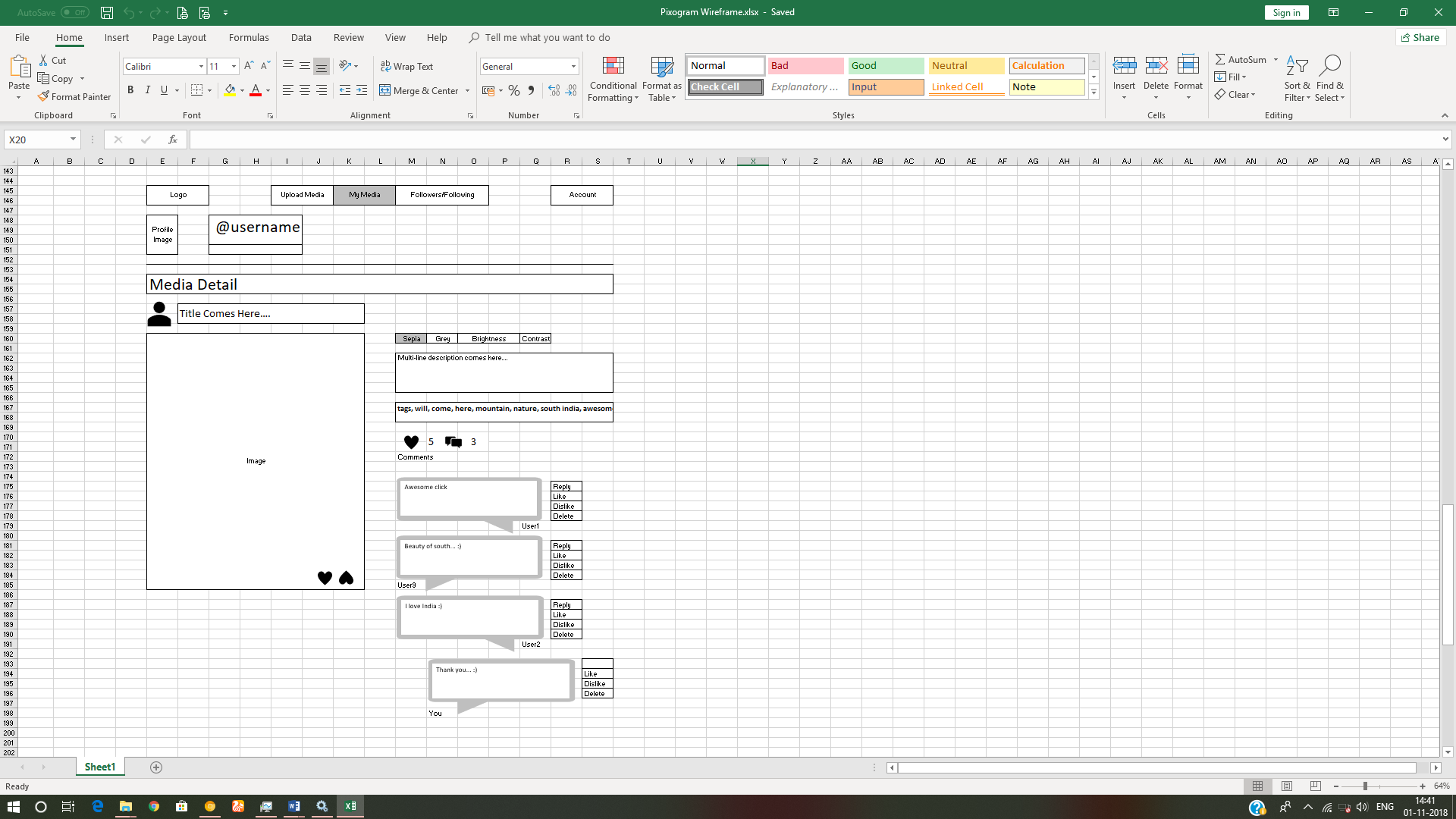


## Media Detail Component

### Media Detail Component Requirement

1. It will display your username on top along with (Follow/Unfollow) toggle button
2. If Image:
   1. Original dimension.
   2. Name of effect applied.
   3. “Make Profile Picture” button, clicking on which will make it a default profile picture for your account. This button is disabled when you are browsing the collection of any other user.
3. If Video:
   1. HTML5 video player
      1. default play/pause/volume button.
      2. video player should also have custom playback progress bar.
      3. Full screen feature
      4. Mute/unmute feature
      5. Replay feature
      6. Loop feature
4. Media title
5. Emoji Icon + number of like. (clickable only once)
6. Emoji Icon + number of unlike. (clickable only once)
7. Emoji Icon + number of comments.
8. Emoji Icon to specify whether it is used for default profile picture.
9. List of comments.
10. Name (hyperlink) of the user who made the comment in front of each comment
11. Link to reply to any comment which will open reply text field.
12. Text field to add new comment to your own post.

### Media Detail Page Wireframe

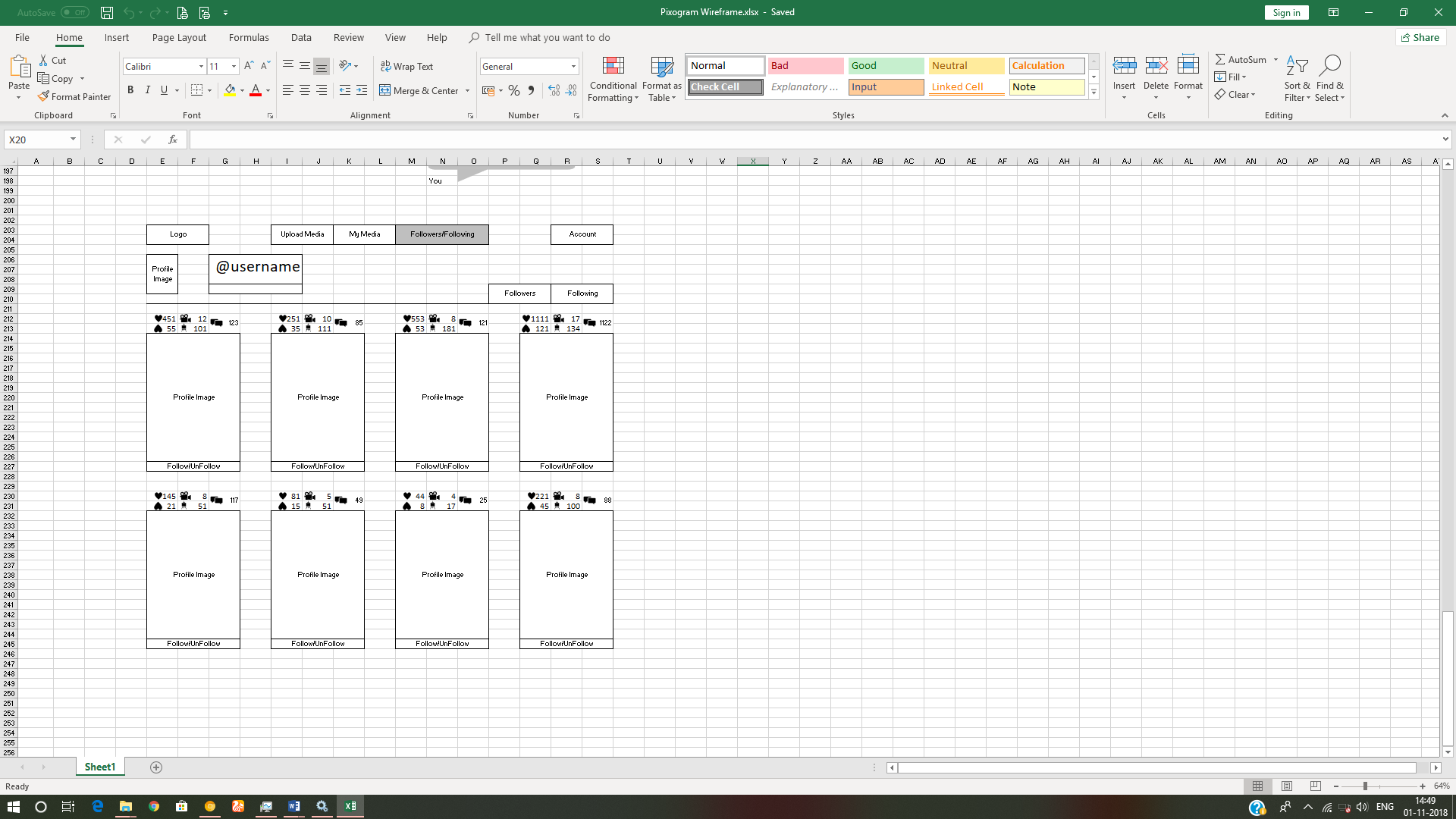


## Followers/Following Component

### Followers/Following Component Requirement

1. Will display the all the followers of your PixoGram account and the users you are following.
2. It will display all followers/following in grid view.
3. Each follower/following profile picture will display below information:
   1. Emoji Icon + total number of like. (not clickable)
   2. Emoji Icon + total number of unlike. (not clickable)
   3. Emoji Icon + total number of comments.
4. User may decide the profiles to be displayed on the page by clicking the buttons on top right:
   1. Followers Button – Will display all followers of your pixogram account
   2. Following Button – Will display all accounts you are following
   3. By default, both buttons are enabled.
5. User can click on any user profile picture and navigate to the “My Media Component” of respective user.
6. Once on the “My Media Component” of the respective user, you can click on any media item to navigate to the respective “Media Detail Component” page.
7. Once on media detail component of the respective user for respective media:
   1. It will display username on top along with (Follow/Unfollow) toggle button
   2. If Image:
      1. Original dimension.
      2. Name of effect applied.
      3. “Make Profile Picture” button is disabled as this media does not belong to your account.
   3. If Video:
      1. HTML5 video player
         1. Default play/pause/volume button.
         2. video player should also have custom playback progress bar.
         3. Full screen feature
         4. Mute/unmute feature
         5. Replay feature
         6. Loop feature
   4. Media title
   5. Emoji Icon + number of like. (clickable only once)
   6. Emoji Icon + number of unlike. (clickable only once)
   7. Emoji Icon + number of comments.
   8. Emoji Icon to specify whether it is used for default profile picture.
   9. List of comments
   10. Name (hyperlink) of the user who made the comment in front of each comment
   11. Link to reply to any comment which will open reply text field.
   12. Text field to add new comment to respective user’s post.

### Followers/Following Component Wireframe



## Account Component

It will consist of 5 sub-component

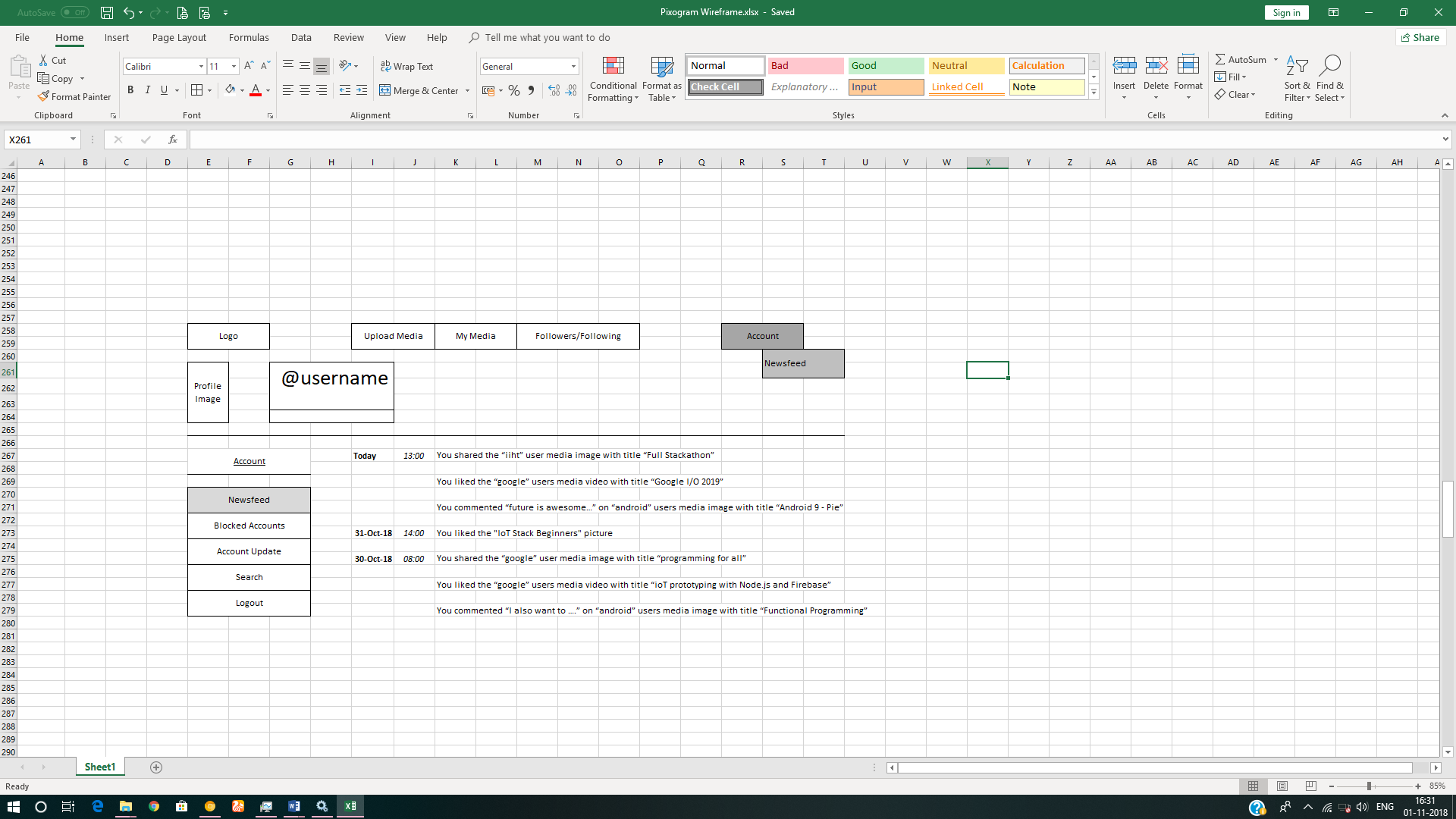
1. Account Details sub-component
2. Activity Log/Newsfeed sub-component
3. Blocked Users sub-component
4. Search sub-component
5. Logout sub-component

### Activity Log/Newsfeed Component

#### Activity Log/Newsfeed Component Requirement

1. Will display the log of all the activity user does on the “PixoGram” app till date.
   1. E.g.
      1. You shared the “iiht” user media image with title “Full Stackathon”
      2. You liked the “google” users media video with title “Google I/O 2019”
      3. You commented “future is awesome…” on “android” users media image with title “Android 9 - Pie”

#### Activity Log/Newsfeed Page Wireframe

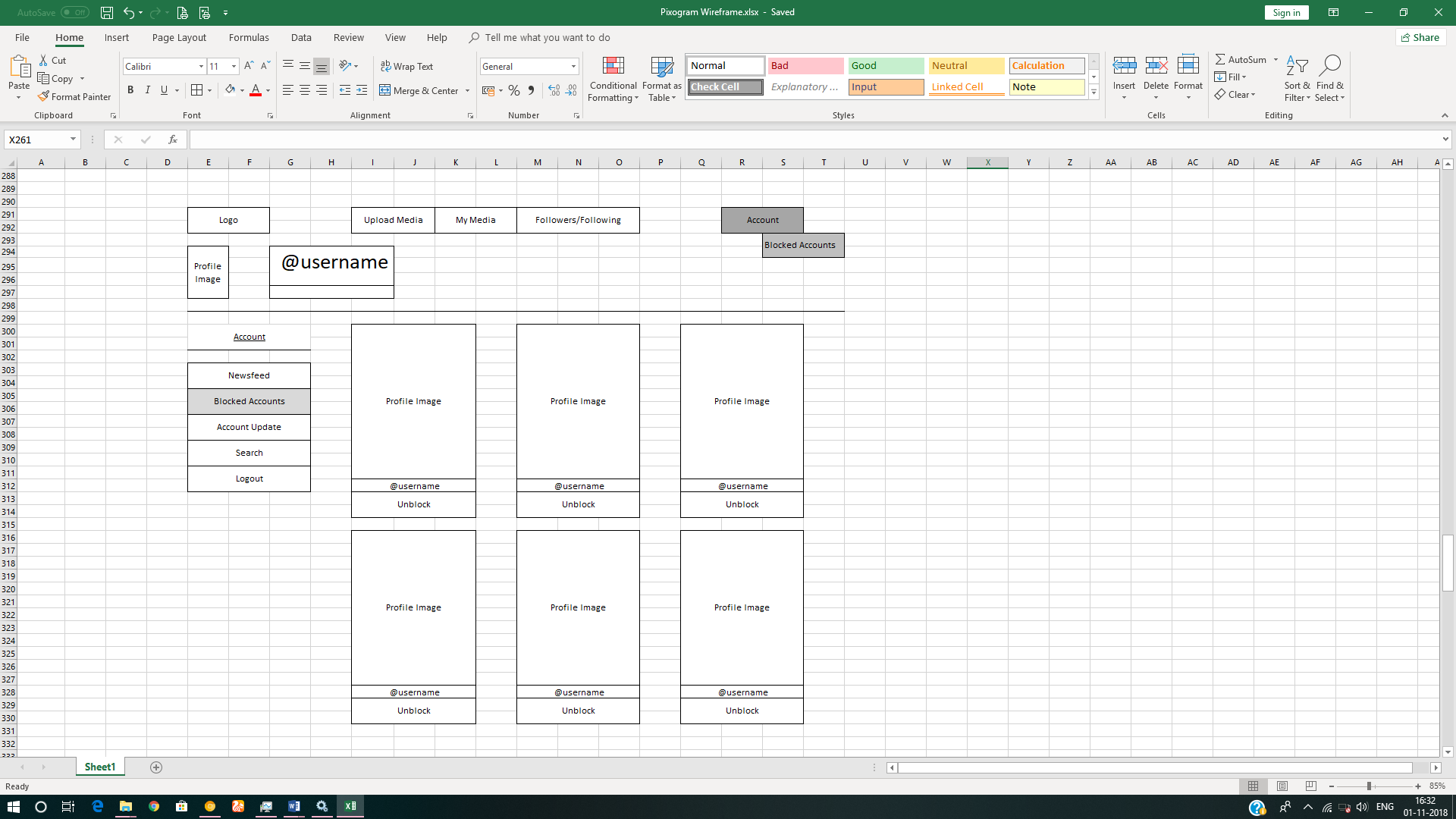


### Blocked Users Component

#### Blocked Users Component Requirement

1. It displays the profile of the accounts who are blocked by you.
2. Blocked accounts cannot view your account on PixoGram.

#### Blocked User Component Wireframe

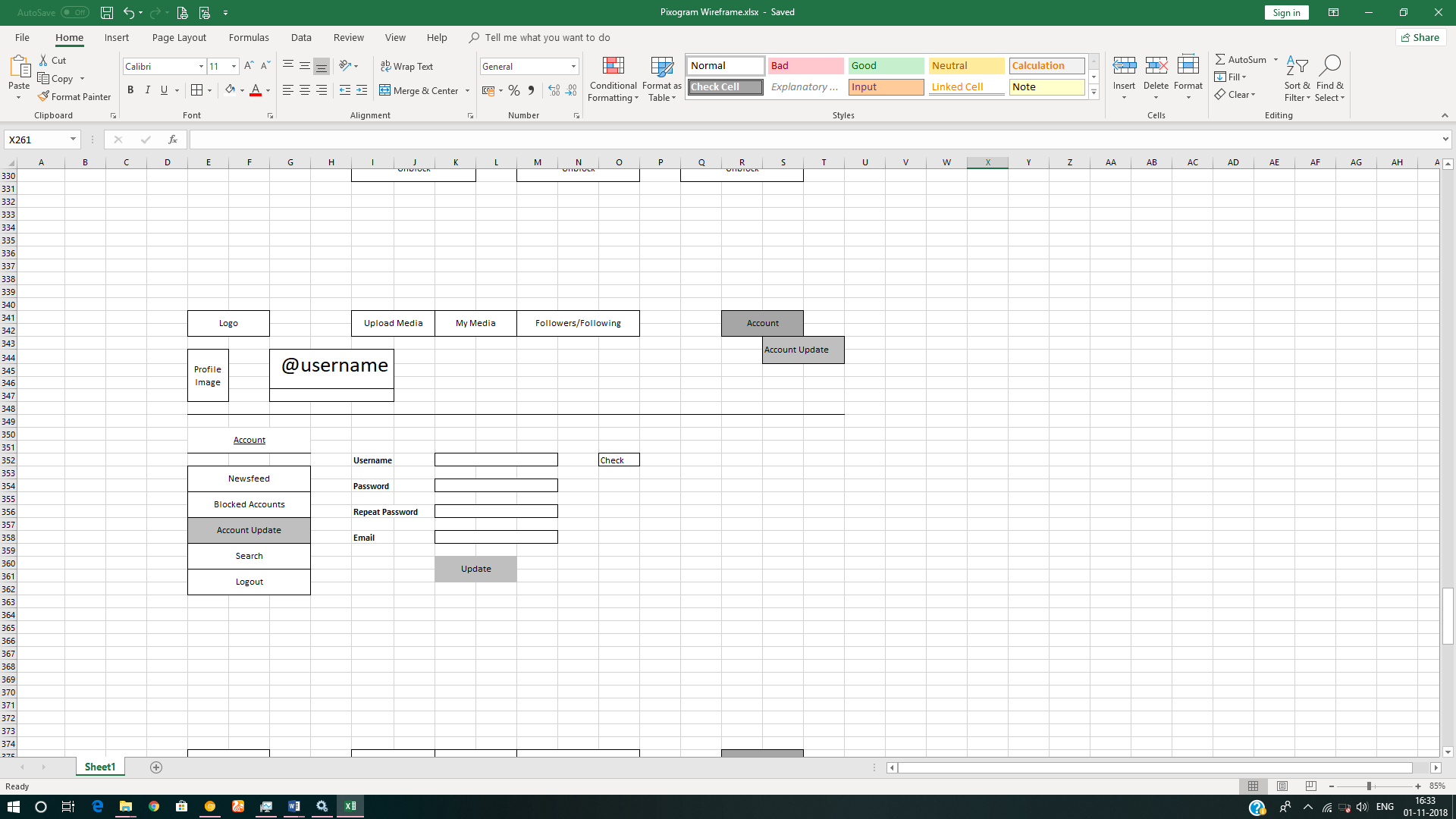


### Account Details Component

#### Account Details Component Requirement

1. It allows you to change the username. Before changing, you need to check if the username is available.
2. You can update email and password.
3. Password validation will follow the same rule as that of password in user registration module.

#### Account Details Component Wireframe

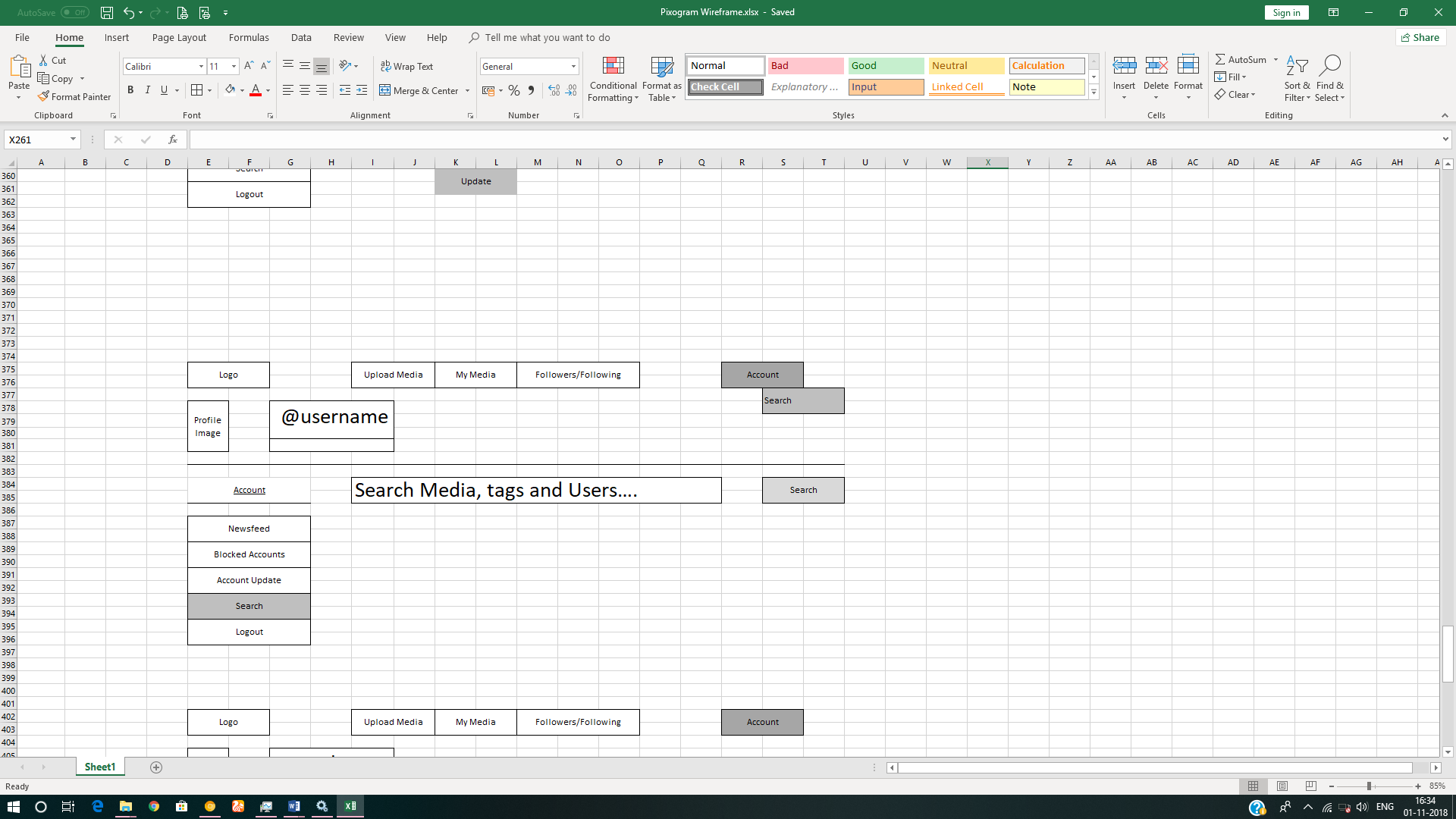


### Search Component

#### Search Component Requirement

* + - 1. User should be able to search content via tags, media title, media description and usernames

#### Search Component Wireframe



# Full Stack Technologies

The technologies included in Full Stack are not limited to following but may consist of:

* UI Layer (HTML5, CSS3, Bootstrap 4, JavaScript, Jquery, Angular 4/6)
* Middleware Restful API (Spring Boot Restful & MicroServices, JAX-RS, Spring MVC)
* Database Persistence ( Hibernate)
* Database layer (MySQL or MongoDB)
* Ancillary skills (GIT, Jenkins(CI/CD), Docker, Maven) etc.

To complete this case study, you should be comfortable with basic single page web application concepts including REST and CRUD. You may use angular-cli to create your template project. All web pages need to be responsive.

Ref1: https://cli.angular.io/

Ref2: https://github.com/angular/angular-cli

# Technical Spec – Solution Development Environment

## Front End Layer

|  |  |
| --- | --- |
| **Framework(s)/SDK/Libraries** | **Version** |
| Angular with TypeScript | 4/6 |
| Bootstrap | 3.0 or above |
| CSS | 3 |
| HTML | 5 |
| JavaScript | 1.8 or above |
| JQuery | 1.3 |

## Middle Tier Layer

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| Java Stack | Spring Boot | 1.5 or above |
| Spring MVC | 4.0 or above |
| JDK | 1.7 or above |
| Maven | 3.x or above |

## Database & Integration Layer

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| Java Stack | Hibernate | 4.0 or above |
| JAX-RS Jersey/ Spring Restful |  |
| MySQL | 5.7.19 |
| MongoDB | MongoDB | 3.4 |
| NoSQL |  |

## Ancillary Layer

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| Source Code Management Tool | GIT | 2.14.2 |
| Build Tool/JAVA Stack | Maven | 3.x |
| Testing Tool/JAVA Stack | JUnit/Mockito | 4.x |
| Testing Tool/JAVA Stack | Spring Test | 4.x |
| Controllers can be tested using Postman Tool | | |

## Security

|  |  |
| --- | --- |
| **Name** | **Version** |
| Spring Boot Security |  |
| JWT |  |

## Deployment & Infrastructure

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| Docker | - |  |
| Apache Tomcat | - |  |
| Jenkins(CI/CD) | - |  |
| Node | - |  |

## Editors

|  |  |
| --- | --- |
| **Name** | **Version** |
| STS(Spring Tool Suite) |  |
| Visual Studio Code |  |

# 

# Deliverables of this Phase

1. Html templates along with Angular components
2. Unit Tested Angular Components using Jasmine

# Important Instructions

1. Consider using below Java features
2. Lambda Expressions
3. Collection Streams
4. Generics
5. Sample Design provided is just for reference, Associates can make changes over it or follow their own Design.
6. Based on your current work, alternate Technologies can be used, for example ReactJS instead of Angular, etc…, however prior approval from the Mentor is required.
7. Please make sure that your code does not have any compilation errors while submitting your case study solution.
8. The final solution should be a zipped code having solution. Solution code will be used to perform Static code evaluation.
9. Implement the code using best design standards/family Design Patterns.
10. Use Internationalization for all the labels and messages in Rest API Development.
11. Do not use System out statements or console.log for logging in Rest API and FrontEnd respectively. Use appropriate logging methods for logging statements/variable/return values.
12. If you are using Spring Restful or Jersey JAX-RS to develop Rest API, then use Maven to build the project and create WAR file.
13. Write web service which takes input and return required details from database.
14. Use JSON format to transfer the results.

For any further queries you can contact fullstack@iiht.com