# Team 18: Project Backlog CS408

## CorkBoard

## Team Members:

Laila Kassar, Nicolas James, Thomas Simons, Quinn Spillane, Vincent Ma

## Problem Statement

Students living in Purdue dorms don't have access to at-a-glance information useful to the entire floor. Weekly newsletters exist, but that information can become quickly outdated. Our program, CorkBoard, provides an information dashboard display that can be displayed near a floor's elevator or common area, with information dynamically updated from multiple sources, at the Floor RA's prerogative.

## Background Information

Getting information out to residents on a dorm floor is often a patchwork of weekly newsletters, wall posters/notices, and GroupMe announcements. With residents coming and going at all hours, and GroupMe not the most stable messaging application, it can be difficult for important information about upcoming events to be disseminated in an effective way. The solution we propose is for a digital signage board intended for placement near elevators or stairs, with constantly updating weather, news, and floor information. This provides residents a useful source of information that is up to date and relevant, available for perusal while waiting for the elevator or before going downstairs. A digital signboard at these bottlenecks through which every student in the hall regularly goes provides the greatest chance of them actually reading it.

## Environment

We intend to use the Visual Studio IDE, with C# as our programming language. The UI will be made using .NET WPF libraries. Weather information will use the API from the national weather service. Information that does not originate from the internet will be stored locally as a configuration file that is read from on startup.

## Functional Requirements

| Backlog ID | Functional Requirement | Hours | Status |
|---|---|---|---|
| 1 | As a user, I want to be able to view announcements. | 5 | Planned for Sprint 2 |

| | | | |
|---|---|---|---|
| 2 | As an RA, I want to be able to add announcements. | 6 | Planned for Sprint 2 |
| 3 | As a user, I want announcements to be updated frequently. | 3 | Planned for Sprint 2 |
| 4 | As a user I want to be able to view the CityBus schedule. | 4 | Planned for Sprint 2 |
| 5 | As a user, I want to be able to view the temperature for the day. | 3 | Planned for Sprint 1 |
| 6 | As a user, I want to be able to view the weather for the day. | 3 | Planned for Sprint 1 |
| 7 | As the application, I want the weather information to update periodically. | 4 | Planned for Sprint 1 |
| 8 | As the application, I want the CityBus schedule to update periodically. | 3 | Planned for Sprint 2 |
| 9 | As a user, I want all information to display on a single screen. | 5 | Planned for Sprint 1 |
| 10 | As a user, I want the current time at a glance. | 2 | Planned for Sprint 1 |
| 11 | As a user, I want the current date at a glance | 2 | Planned for Sprint 1 |
| 12 | As a user, I want to see the top news headlines. | 4 | Planned for Sprint 2 |
| 13 | As the application, I want to be able to be self sustaining and not crash after long periods of time.. | 5 | Planned for Sprint 1 |
| 14 | As an RA, I want some level of customization over the dashboard's layout. | 4 | Planned for Sprint 1 |
| 15 | As an RA, I want an obvious error message display if any information cannot be updated. | 2 | Planned for Sprint 1 |
| 16 | As an RA, I want to remove old announcements | 3 | Planned for Sprint 2 |
| 17 | As an RA, I want to edit announcements | 3 | Planned for Sprint 2 |
| 18 | As the application, I will update the news periodically | 4 | Planned for Sprint 2 |
| 19 | As an RA, I want to choose the color of the board | 3 | Planned for Sprint 1 |

| | | | |
|---|---|---|---|
| 20 | As an RA, I want to be able to import/export settings | 5 | Planned for Sprint 2 |
| 21 | As a user I want to be able to see the most recent tweets from the Purdue Alerts Twitter account | 5 | Planned for Sprint 2 |
| 22 | As a user I want to view an image displayed on the board. | 2 | Planned for Sprint 1 |
| 23 | As an RA I want to be able to choose an image to be visible on the board | 3 | Planned for Sprint 1 |
| 24 | As the application, I will get active weather alerts. | 3 | Planned for Sprint 1 |
| 25 | As an RA, I want to change the refresh rate | 3 | Planned for Sprint 1 |
| 26 | As an RA, I want to add a URL to get announcements from. | 2 | Planned for Sprint 2 |
| 27 | As an RA, I want to change the location weather information is pulled from | 2 | Planned for Sprint 1 |
| 28 | As an RA, I want to set the twitter feed | 3 | Planned for Sprint 2 |
| 29 | As an RA, I want to set the bus schedule | 2 | Planned for Sprint 2 |

# Non-Functional Requirements

## Architecture

We intend for our app to be compatible with any modern Windows 10 PC with some form of graphics processor. Internet access is required in order to properly display live data, but beyond the config file used at startup, no persistent data is stored on the machine.

## Security

Because users don't interact with the board beyond reading information off of it, there's no risk of sensitive user data being compromised, because none exists nor is stored. Although the potential exists for vandalism of the board or its contents, the risk is no greater (and is arguably more difficult) than the existing vandalism risk of a student printing out fake newsletters and posters and posting them on the dorm floor.

### Usability

As a baseline, the display should be simple and human-readable, with information displayed clearly and concisely. The configuration file for making changes should be well-commented and walk the RA through configuring the display properly.

### Performance

Because the display uses very few dynamic components and only updates information periodically, no performance issues are expected, considering the display software monopolizes use of the host computer's monitor and no other software is expected to be run on the machine at the same time. Network traffic is minimal and should be quick to update even on a slow connection - The XML Weather API and RSS are both lightweight, time-tested systems.

# Use Cases

## Case 1:

View announcements

| Action | System Response |
|---|---|
| 1. Launch Application | 2. Connects to internet |
| | 3. Pull announcement text file |
| | 4. Read file |
| | 5. Display announcements |

## Case 2:

Add announcements

| Action | System Response |
|---|---|
| 1. RA navigates to URL | |
| 2. RA uploads text file with announcements | 3. Text file is uploaded and stored on server to be pulled later by the application |

## Case 3:

Refresh announcements

| Action | System Response |
|---|---|
| 1. Timed trigger is triggered | 2. Connect to internet |
| | 3. Pull text file from URL |
| | 4. Read text file |
| | 5. Update announcement display |

## Case 4:

View CityBus schedule

| Action | System Response |
|---|---|
| 1. Launch application | 2. Connects to internet |
| | 3. Pull bus schedule from CityBus site |
| | 4. Parse schedule as necessary |
| | 5. Display bush schedule |

## Case 5:

View Temperature

| Action | System Response |
|---|---|
| 1. Refresh trigger is triggered | 2. Connects to internet |
| | 3. Pulls weather feed from weather.gov API |
| | 4. Read weather feed |
| | 5. Update temperature display |

## Case 6:

View weather for the day

| Action | System Response |
|---|---|
| 1. Launch application | 2. Connect to internet |
| | 3. Pull weather data from government site |
| | 4. Format response |
| | 5. Display weather |

## Case 7:

Application update weather periodically

| Action | System Response |
|---|---|
| 1. Refresh triggers | 2. Connect to internet if not already |
| | 3. Pull weather data from National Weather Service |
| | 4. Format data as necessary |
| | 5. Display weather |

## Case 8:

Update CityBus schedule periodically

| Action | System Response |
|---|---|
| 1. Refresh triggers | 2. Connect to internet if not already |
| | 3. Pull bus schedule from CityBus site |
| | 4. Format data as necessary |
| | 5. Display schedule |

## Case 9:

Display all data on a single screen

| Action | System Response |
|---|---|
| 1. Application launches | 2. Connect to internet |
| | 3. Pull all necessary data from all sources |
| | 4. Format all data as necessary |
| | 5. Display all data in easy to read fashion |

## Case 10:

Time is available at a glance

| Action | System Response |
|---|---|
| 1. Application lanches | 2. Time is retrieved from system |
| | 3. Time is displayed |
| 4. Resident glances at screen and sees the time | |

## Case 11:

Date is available at a glance

| Action | System Response |
|---|---|
| 1. Application lanches | 2. Date is retrieved from system |
| | 3. Date is displayed |
| 4. Resident glances at screen and sees the date | |

## Case 12:

Top news headlines are displayed

| Action | System Response |
|---|---|
| 1. Application launches | 2. Retrieve new stories from news source |
| | 3. Get headlines of top stories |
| | 4. Display headlines retrieved |

## Case 13:

App does not crash after long periods of time

| Action | System Response |
|---|---|
| 1. Application launches | 2. Startup processes happen |
| | 3. Application stays open for a long period of time |
| | 4. Application doesn’t not crash |

## Case 14:

RA layout customization

| Action | System Response |
|---|---|
| 1. RA goes to settings | 2. Settings view is launched |
| 3. RA navigates to options for layout | 4. Options for layouts displayed |
| 5. RA selects option | 6. Changes to layout are displayed |

## Case 15:

Display error message when information can’t be updated

| Action | System Response |
|---|---|
| 1. Refresh triggers | 2. Request for new data is sent out |
| | 3. Some sort of error is received for one or more of the data sources |
| | 4. Error message is generated based on which sources failed |
| | 5. Error message displayed |

## Case 16:

Remove old announcements

| Action | System Response |
|---|---|
| 1. RA visits URL where announcements are uploaded. | |
| 2. RA deletes old announcement files | 3. Selected announcement files deleted |
| | 4. Old announcements no longer viewable |

## Case 17:

Edit announcements

| Action | System Response |
|---|---|
| 1. RA visits URL where announcements are uploaded | |
| 2. RA chooses announcement to update | |
| 3. RA uploads updated announcement | 4. Selected announcement file is replaced |
| | 5. Updated announcement is shown |

## Case 18:

Update new periodically

| Action | System Response |
|---|---|
| 1. Refresh triggers | 2. News stories are pulled from news source |
| | 3. Headlines are obtained and formatted |
| | 4. Headlines are displayed |

## Case 19:

Color customization

| Action | System Response |
|---|---|
| 1. RA opens settings | 2. Settings view is launched |
| 3. RA navigates to color customization | 4. Color picker is displayed |
| 5. RA chooses desired color for board | 6. Display color of board is changed |

## Case 20:

Import/export settings

Import

| Action | System Response |
|---|---|
| 1. RA navigates to settings | 2. Settings menu launched |
| 3. RA selects import settings | 4. Open file chooser prompt |
| 5. RA chooses settings file | 6. Settings imported from file |

Export

| Action | System Response |
|---|---|
| 1. RA navigates to settings | 2. Settings menu launched |
| 3. RA selects export settings | 4. Settings are exported to file |
| | 5. File saved to file system |

## Case 21:

See tweets from Purdue Alerts Twitter account

| Action | System Response |
|---|---|
| 1. Application launched | 2. Call Twitter API to get tweets from Purdue Alerts |
| | 3. Display most recent tweets received from API call |

## Case 22:

Display image on board

| Action | System Response |
|---|---|
| 1. Application launched | 2. Default image or chosen image displayed |

## Case 23:

RA chooses image to display on board

| Action | System Response |
|---|---|
| 1. RA goes to settings | 2. Settings menu launched |
| 3. RA selects to show custom image | 4. File chooser opened |
| 5. RA chooses image | 6. Image chosen is displayed instead of old image |

## Case 24:

Get any active weather alerts

| Action | System Response |
|---|---|
| 1. Application launches | 2. Pull weather data |
| | 3. Check for active weather alerts |
| | 4. If there are active weather alerts, display them |

## Case 25:

Change the refresh rate

| Action | System Response |
|---|---|
| 1. RA selects settings | 2. Settings view is launched |
| 3. RA selects refresh rate | |
| 4. RA inputs new refresh rate | 5. Refresh rate for data pull is updated |

## Case 26:

Add URL to pull announcements from

| Action | System Response |
|---|---|
| 1. RA selects settings | 2. Settings view is launched |
| 3. RA enters new URL to pull announcements from | 4. Announcements URL is updated |

## Case 27:

Change location weather is pulled from

| Action | System Response |
|---|---|
| 1. RA selects settings | 2. Settings view is launched |
| 3. RA enters new location for weather | 4. Location that is used for weather data pulls is updated |

## Case 28:

As an RA, I want to set the twitter feed

| Action | System Response |
|---|---|
| 1. RA selects settings | 2. Settings view is launched |
| 3. RA enters twitter url and account details | 4. Info that is used for twitter data checks is set |

## Case 29:

As an RA, I want to set the bus schedule

| Action | System Response |
|---|---|
| 1. RA selects settings | 2. Settings view is launched |
| 3. RA enters url location for bus schedules and preferred stop location. | 4. Bus schedule is updated with newest arrivals for that stop. |