

Kevin Ehresman

CSC-415 Assignment 3 Analysis and Design

Dr. Pulimood

Pathname to project repo on Github: <https://github.com/ehresmk/MyHometown>

(Developing a mobile app on my personal computer not on the VM)

1)Diagrams & Descriptions (Diagrams in second PDF)

-Detailed Use-Case Descriptions

Use case: Create a city page

Primary actor: Admin

Goal in context: To create a page where subscribers can view and receive alerts from the city/town that the page is associated with.

Preconditions: A page for the desired city does not already exist.

Trigger: The admin clicks the “Create City Page” button.

Scenario:

1. Admin: Clicks the “Create City Page” button.
2. Admin: Enters the name of the city they wish to add a page for.
3. Admin: Observes if the system finds a page for that city.
4. Admin: Asked to confirm the city they want to create a page for.
5. Admin: Observes a successful creation of a page for the city entered.

Exceptions:

1. A page for the desired city already exists: The admin will be prompted to return to the main menu, or enter a different city to create a page for.

Use case: Post alerts on a city page

Primary actor: Admin

Goal in context: To post an alert on a page for a specific city. (A subscriber will receive a notification depending on whether or not they have that enabled)

Preconditions: A page for the desired city must already exist.

Trigger: The admin clicks “Create Alert” on the city page.

Scenario:

1. Admin: Clicks the “Create Alert” button on a city page.
2. Admin: Inputs the type of alert they are creating.
3. Admin: Inputs the title of the alert.
4. Admin: Inputs the description of the event/warning/situation/etc
5. Admin: Reviews entered information.
6. Admin: Clicks the “Send Alert” button.
7. Admin: Observes that the alert has been sent.

Exceptions:

1. The Admin attempts to send an alert with no information: The system will ask the admin to input information for the alert before sending it.

Use case: View posts on a city page

Primary actor: Subscriber/Admin

Goal in context: To view alerts recently posted by an admin on a specific city’s page.

Preconditions: The subscriber must be subscribed to the city in order to view previously sent alerts on that city’s page. The admin must be an administrator of that particular city’s page in order to view past alerts. Both the admin and subscriber must be logged in.

Trigger: When the subscriber logs in they will immediately be transferred to this page. The admin must click “View Alerts” from their homepage.

Scenario:

1. Subscriber: Is transferred from the login screen to their main city page.
2. Subscriber: Observes previous alerts and can scroll through them and read the detailed descriptions of the alerts.

Exceptions:

1. The subscriber is not subscribed to any pages: There will be a “You are not subscribed to any pages” prompt and a search bar for the subscriber to search for a city page to subscribe to. (The search bar is if time permits, only 1 city page will be implemented for the initial build of the application)

Secondary Actors: Admin

1. Admin: Clicks the “View Alerts” button from their homepage.
2. Admin: Observes alerts previously sent to the subscribers of the specific city page.

2)User Interfaces

The image displays four sequential mockups of the MyHometown app's user interface:

- Mockup 1 (Login):** Features the title "MyHometown", a welcome message, and fields for "Username" and "Password". It includes a "Login" button and a "Create an Account" button at the bottom.
- Mockup 2 (Account Creation):** Features the title "MyHometown", a prompt to enter username and password, and fields for "Username", "Password", and "Password Again". It includes "Create Account" and "Back" buttons.
- Mockup 3 (City Selection):** Features the title "MyHometown" and two city selection boxes: "New Brunswick" and "Milltown", each with a "Recent Alerts:" label. It includes a "Search" field, a "Logout" button, and a "Back" button at the bottom.
- Mockup 4 (City Details):** Features the title "MyHometown" and a list of alerts for "New Brunswick". The alerts include details like "Home robbery at 64 Harrison Ave.", "Posted: 3/25/2020", and "Description:". It includes a "Back" button at the bottom.

The above pictures depict my mockups for the UI of the MyHometown app I am developing. All the controls are visible to the user, and in the example of the first two pictures (the login and account creation pages) give prompt feedback to the user if they input an incorrect username/password or if they leave entries blank. All the buttons and input boxes are labelled according to their functionality. In the third picture, the user can click on those boxes for each city to go directly to the page associated with that city to see all recently posted alerts.

Each user interface presents information in the appropriate context that is consistent. When a user clicks the create an account button, it brings them to the appropriate screen in order to create their account. Frequent users will be presented with the third screen showing them the city pages that they are subscribed to, so they don't have to search for them every time they open the app. The user is given prompt and useful feedback when they input an incorrect username/password, or if they search for a city which does not yet have a page (if time permits the search functionality will be implemented).

The changing of one UI screen to the next is how the application yields closure to the user. When the user successfully logs in they will be brought to their home screen (third picture), where they can see all the city pages they are subscribed to. When a user successfully searches for a city, they will be directed immediately to the city page for that searched city. For my UI, error handling is rather simple, when a user inputs an incorrect username/password it displays an error message right at the bottom of the screen and allows the user to reinput their credentials. On both the second and fourth UI mockup pictures, a "back button" can be seen in the bottom right corner of the screen. This allows the user to easily reverse their actions and return to the previous page which in the case of the second screen returns them to the login screen, and for the fourth screen it returns them to the homepage (third screen).

The user is always in control during their usage of MyHometown. When they login they can see the list of city pages that they're subscribed to and can click on whichever one they want in order to see a list of recent alerts that were sent on that page. The memory load placed on the user in this application is very minimal, they really only need to remember what cities they want to receive alerts from so that they can subscribe to those pages.

3)Consideration to good programming concepts and techniques

My project has a number of modules to handle user login, account creation, and user logout that have already been implemented. I plan to implement separate modules for the searching of a city page and the loading of city pages for a specific subscriber. The use of these modules allows for the reusability of certain parts of my application. Perhaps in the future I'm designing another mobile application that isn't related to cities or hometowns, but I would like to use a login/logout system that I have already developed. Since these modules are independent of the code relating to the city pages, I can easily reuse these modules when developing a new application.

So far I haven't implemented many algorithms, just the storing and processing of user input through the login system. It is important to take into consideration however, as writing elegant code makes it easier to explain to a stakeholder or non-software engineer. Making sure the algorithms are efficient is important for the user as well. You don't want an application that takes 5 minutes to process the user logging into the system, convenience and efficiency are key when developing a mobile application that users will actually want to utilize.

At this point in development I haven't implemented any data structures but I plan on using an array for storing the city pages that subscribers are subscribed to. I decided to use an array because it allows for easy access and I don't need to order the data in the array in any specific way. Perhaps if I wanted to implement a "primary city" functionality where that city page is always shown at the top of the list for the subscriber, then I could use a single variable for said city but I think an array is an appropriate data structure for storing the list of subscribed cities.

-Test Case Design

Within Android Studio there is a built in debugger and other tools that identify syntactically incorrect code, so I will (and have been) utilizing those tools during the course of development. Below are a number of test cases I have designed in order to test the functionality of my application:

Functionality Tested	Inputs	Expected Output	Actual Output
Create a city page	1)-Login as Admin -Click "Create City Page" button -Enter a city name (New Brunswick) 2)-Login as Admin -Click "Create City Page" button -Enter a city name that already has a page (New Brunswick)	1)City page successfully created, and display the created page. 2)City page could not be created due to an already existing page, and prompts the user to try a different city.	N/A
Post alerts	1)-Login as Admin -Navigate to the desired created city -Click the "Post Alert" button -Enter a title, and message for the alert -Click "Send Alert" 2)(Same scenario as above except...) -Leave the title and message boxes blank -Click "Send Alert"	1)Alert sent successfully, with a display of the alert that was sent with the title and description attached. 2)Could not send an alert because of invalid entries: Please enter a title and description.	N/A
View alerts	1)-Login as subscriber -Select any city page -Observe the alerts that have been posted	1)The alerts that have been posted for the selected city page should be displayed in order of most to least recent.	N/A