

The following document contains descriptions (user stories) for 3 iteration which were worked upon using Scrum/Agile development method.

# Iteration 1 User Stories

As a user of the application, ...

(Approximate point values indicated in [square brackets])

## 1. Display list of all restaurants

- I want the app to be installed with some restaurant and inspection report data for Surrey.
- [5] I want to see a list of all restaurants sorted alphabetically so I can find a restaurant I am interested in.
- [25] I want each item in the list of restaurants to clearly show me some brief information about the restaurant so I can see info at a glance.  
Each restaurant in the list must include:
  - Restaurant name
  - An icon for the restaurant (generic OK)
  - Info on most the restaurant's most recent inspection report:
    - # issues found (sum of crit and non-crit issues)
    - colour for hazard level (low, medium, high)
    - icon for the hazard level
    - how long ago was the inspection done (see next point)
- [5] I want to be told when something happened in an intelligent format so that it's easier to understand than dates:
  - If it was within 30 days, tell me the number of days ago it was (such as "24 days")
  - Otherwise, if it was less than a year ago, tell me the month and day (such as "May 12")
  - Otherwise, tell me just the month and year (such as "May 2018")
- I want to be able to click a restaurant in the list and have it show me the details about that restaurant (see below).

## 2. Display details of single restaurant

- I want to see a screen showing the details about a single restaurant so I can understand the history of inspections at that restaurant.
- [5] I want the restaurant details screen to include:
  - Restaurant name
  - Restaurant address
  - Restaurant GPS coords

- I want it to list all inspections (in a scrollable list), with the most recent at the top, so I can easily understand the timeline of inspections.
- [15] I want each listing for an inspection report to show:
  - # critical issues found
  - # non-critical issues found
  - How long ago the inspection occurred (see formatting notes from above)
  - Change the colour, and display an icon, for the hazard level

### 3. Display details of single inspection

- I want to see the details of a single inspection report to learn all I can about what the inspection found.
- [10] I want the inspection details screen to show:
  - Full date of the inspection (such as "May 12, 2019")
  - Inspection type (routine / follow-up)
  - # critical issues found
  - # non-critical issues found
  - Hazard level icon, hazard level in words, and colour coded.
  - Scrollable list of violations
- [15] I want each violation in the list to show:
  - An icon reflecting the nature of the violation (food, pest, equipment, ...)
  - A brief description of the violation which fits on one line
  - The severity of the violation: Use an icon and colour to make it easy to see if it's critical or not-critical
- I want to be able to tap on a single violation to see the long description. It's OK if this is shown on the screen for a moment (such as a toast or snackbar).

### General

- Use the singleton pattern to allow all parts of the UI to access the model.
- [10] Overall
  - Add a back button on the tool-bar at top when appropriate.
  - From any screen, the Android back button must do the "reasonable" thing.
  - Application flows smoothly.
  - Nice UI layouts; resizes to different size screens reasonably well.
- [0] Must have good quality code (Up to 20 point deduction)
  - Good class, method, variable names.
  - Perfectly formatted code.
  - Comments on all classes (not needed *inside* classes, just on the class level)
  - All strings which end user may see on UI (not LogCat) are in strings.xml
  - UI must be in a separate package from the data "model".
- [10] Correct use of Git and GitLab

- Using GitLab issues to track features to work on.
- Use Git/GitLab branches and GitLab merge requests
- Correctly builds when cloned from GitLab.

## Iteration 2 User Stories

As a user of the application, ...

(Approximate point values indicated in [square brackets])

### 1. [30] Get updated data

- When the app start up, if it's been 20 hours or more since data was last updated, I want the application to check with the City of Surrey's server to see if there is more recent data to be downloaded.
- If there is new data on the server, I want the app to ask the user if they want to update the data so they can choose if now is a good time to do the update.
  - If the user does not want the data updated, then the next time the app starts up it should ask the user again if they want to update.
  - The app must only download the data if an update is available otherwise it will waste the user's bandwidth and battery.
- I want the application to store downloaded data locally so that it can work offline.
- I want the application to initially install with the small data set (from iteration 1).
- Show a please-wait dialog, or a progress dialog, while data is downloading/updating.
  - I need the dialog to have some animation showing the application is working.
  - Plus it must have some way for the user to cancel the download.
  - When the download is cancelled the previously download data should still be used by the app.

### 2. [30] Map

- When the application starts up, I want the default view to be a map centred on the user's current location.
- The map should display pegs showing the location of each restaurant we have data for.
- I want the map to allow the user to pan (move map around) and pinch to zoom.
- I want the restaurant pegs to show the hazard level of the most recent inspection report for a restaurant.
  - Each peg must show the hazard level using a colour and an icon. i.e., icon for medium must be different than high hazard.

- When there are too many pegs in an area, I want the pegs to be clustered intelligently.
- I want the app to show the user's current GPS position on the map as some form of dot or icon which is distinct from the restaurant pegs.
  - As the user moves, I want the dot on the screen to update to a new location, and have the map follow the user so the display stays relevant as the user moves through the city.
- I want the user to be able to interact with a peg to get more information.
  - Tapping on a peg should show a small pop-up display of the restaurant name, address, and hazard level of its most recent inspection.
  - The pop-up display must also allow the user to tap again to go to restaurant's full information screen
- I want the user to have a clear way to toggle between the map screen and the list of restaurants screen (iteration 1) so the user can find a restaurant by either its location on the map, or the list of restaurants.

### 3. [15] Custom images / icons

- I want the icon/image for certain restaurants to be specific to that business so that the user can quickly identify business.
  - At least 10 restaurants must have unique icons in the restaurant list screen. Suggested that these icons be the company's logo.
  - At least 5 of these restaurants should have 4 or more locations in Surrey so that the custom icons come up more frequently in the list.

### 4. [15] Back-button behaviour

- Toggling between map and restaurant list view is independent of the back button behaviour
  - If a user goes from the map screen to the restaurant list screen and then pressing Android's back button, it exits the application.
  - If a user goes from the restaurant list screen to the map screen and then pressing Android's back button, it exits the application.
- Pressing a back button (either Android's or the one on the app's screen) from the single restaurant screen takes the user back to either the map screen or the restaurant list screen, whichever the user last visited.
- On the single restaurant screen, tapping the GPS coords on a restaurant closes the current screen and returns to the map screen, selecting that restaurant and showing the small pop-up info about the restaurant.

## General

- [10] Overall

- Nice UI layouts; resizes to different size screens reasonably well.
  - All strings which end user may see on UI (not LogCat) are in strings.xml
  - Add a back button on the tool-bar at top when appropriate.
  - From any screen, the Android back button must do the "reasonable" thing.
  - Application flows smoothly.
- [0] Must have good quality code (Up to 20 point deduction)
  - Good class, method, variable names.
  - Perfectly formatted code.
  - Comments on all classes (not needed *inside* classes, just on the class level)
  - UI must be in a separate package from the data "model".
- [10] Correct use of Git and GitLab
  - Using GitLab issues to track features to work on.
  - Use Git/GitLab branches and GitLab merge requests
  - Correctly builds when cloned from GitLab.
  - Evidence of some use of code reviews

## Iteration 3 User Stories

As a user of the application, ...

(Approximate point values indicated in [square brackets])

### 1. [40] Search / Filter

- I want to be able to search on the map screen for restaurants of interest so I can limit the markers I see on the map.
- I want to be able to search on the restaurants list page for restaurants of interest so I can limit the restaurants I see in the list.
- Searching details
  - I want to be able to clear the current search to resume seeing all restaurants.
  - I want to preserve the current search while switching between the map and restaurants list. For example, if I searched for "pizza" on the map, when I switch to the restaurant list I want to still be seeing search results for "pizza".
  - If I search for "pizza" and then tap on a restaurant to view that restaurant's details screen (from either the map or the restaurant list), when I select back on the restaurant details screen I want to return to the previous screen with the search results intact.
- When I search, I want to be able to see only restaurants...

- whose name matches something I type in (so if searching for "pizza" I get "Boston Pizza #425", "MrPizzas", etc)
  - whose most recent inspection had a specific hazard level
  - which has had  $\leq$  (or  $\geq$ ) N critical violations within the last year
  - which have been tagged as a favourite (see below)
- I want to be able to combine the above criteria, such as: Show me all my favourite restaurants which have "bar" in the name, with a most recent inspection hazard level of low, and which had 5 or less critical violations within the last year.
  - When combining multiple criteria, I want to only see restaurants which satisfy all criteria.
  - I don't need search to support boolean operators.
  - I am OK with it searching for the full text which I type in. For example, if I search of "pub bar" I want to see "Pub barnacle" and not "The bar-none best pub"

## 2. [20] Favourites

- From the restaurant details screen, I want to be able to mark (or un-mark) the restaurant as a favourite.
- In the restaurants list I want my favourite restaurants to be noticeable. For example, have an extra icon overlaid on their image, or a different background or border colour (though this is very flexible). As I scroll through my list, these should stand out to me.
- When new data is downloaded from the City of Surrey's server, I want to be told if there are any new inspections for my favourite restaurants.
  - For example, after the app has updated the data, display a list of my favourite restaurants which have new inspections.
  - When being told about a new inspection on my favourite restaurants, I want to be told at least the name of the restaurant, the date of the most recent inspection, and the hazard level of the newest inspection.

## 3. [30] Internationalize

- I want the app to support more than one language.
- When the application starts, if it supports the language my Android device is set to, I want it to show me everything in that language.
  - If the application does not support the language of my device, default to English.
- All user facing strings should be translated, with the possible exception of:
  - Data read out of the data files (like those downloaded when updating)
  - Networking errors (such as "file not found"...) if they are generated by the networking package

- Don't forget to translate the brief one-line summaries of the violations shown in the violations list.
- Technical notes: Your `docs/readme3.txt` file must list the languages you support. You pick the language to support; I suggest only languages which are drawn left to right like English to reduce your layout complexity.

## General

- [0] Must have good quality code (Up to 20 point deduction)
  - Good class, method, variable names.
  - Perfectly formatted code.
  - Comments on all classes (not needed *inside* classes, just on the class level)
  - UI must be in a separate package from the data "model".
- [10] Correct use of Git and GitLab
  - Using GitLab issues to track features to work on.
  - Use Git/GitLab branches and GitLab merge requests
  - Correctly builds when cloned from GitLab.
  - Evidence of some use of code reviews