Course Project Documentation

Restaurant Health Inspection App

October 29, 2020

Table of Contents

1. Introduction

1a Motivation

With the current pandemic of Covid-19, it is important to know what restaurants are safe to eat from. For delivery, the food should be safely made with standard washing procedures. For eating in, the tables should be distanced from each other, to reduce the spread of Covid-19.

1b Task Description

Create a Restaurant Health Inspection App for Android. The app must receive data from two csv files with restaurant details and inspection report form Surrey. All user requirements must be fulfilled to move on to the next iteration.

1c Objective

The Android app must be simple to use and easy to navigate. The code must be in good quality, with object-oriented design. Teammates will understand the methodology of Agile and Scrum

2. Problem Analysis

2a User Requirements

General

The good quality code must use singleton pattern to access the model in all user interface. The application must have back button, nice UI layouts, smooth application. Correct use of Git and GitLab is necessary.

Iteration 1

Screen 1: List of all restaurants

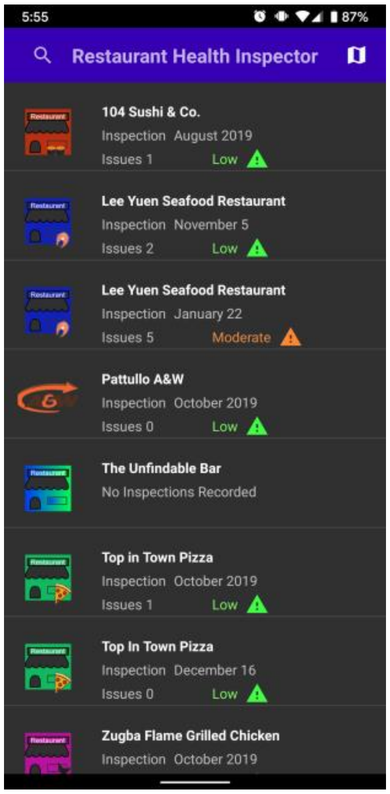
- App must be installed with restaurant and inspection report data from Surrey
- The list of restaurants must be in alphabetical order
- Each restaurant must have brief information including:
    - Restaurant name and icon
    - Information on most recent inspection report including:
        - Number of issues found
        - Hazard colour and icon
        - Date of inspection (using intelligent date)
- Intelligent date format:
    - If within 30 days, tell number of days ago (24 days)
    - If within a year, tell month and day (Oct 29)
    - Else, tell month and year (Oct 2020)
- Once restaurant is clicked, it must show details of the restaurant

Screen 2: Details of single restaurant

- Screen shows restaurant details including:
  - Restaurant name, address, GPS coordinates
- List all inspections, sorting by the latest
- Each inspection report must have brief information including:
  - Number of critical and non-critical issues
  - Date of inspection (using intelligent date)
  - Hazard colour and icon
- Once inspection is clicked, it must show details of inspection

Screen 3: Details of single inspection

- Screen shows inspection details including:
  - Inspection full date and type
  - Number of critical and non-critical issues
  - Hazard level icon and colour
  - List of violations which includes:
    - Nature of violation icon
    - Description of violation
    - Severity of violation
- Once violation is clicked, a long description must temporally pop up



5:55 87%

Restaurant Health Inspector

104 Sushi & Co.
Inspection August 2019
Issues 1 Low

Lee Yuen Seafood Restaurant
Inspection November 5
Issues 2 Low

Lee Yuen Seafood Restaurant
Inspection January 22
Issues 5 Moderate

Pattullo A&W
Inspection October 2019
Issues 0 Low

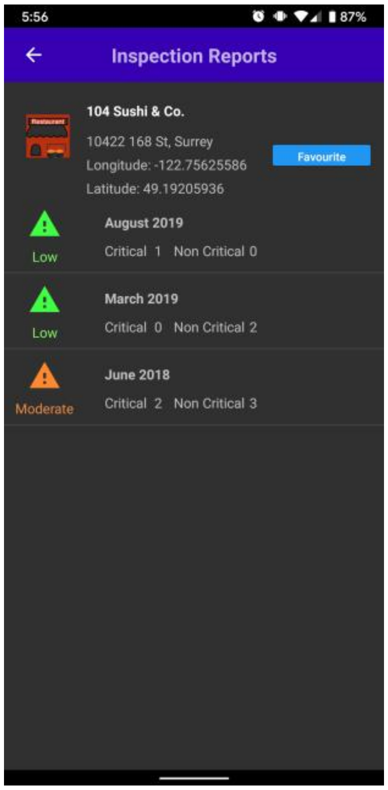
The Unfindable Bar
No Inspections Recorded

Top in Town Pizza
Inspection October 2019
Issues 1 Low

Top In Town Pizza
Inspection December 16
Issues 0 Low

Zugba Flame Grilled Chicken
Inspection October 2019

Figure 1 (Screen 1)



5:56 87%

Inspection Reports

104 Sushi & Co.
10422 168 St, Surrey
Longitude: -122.75625586
Latitude: 49.19205936

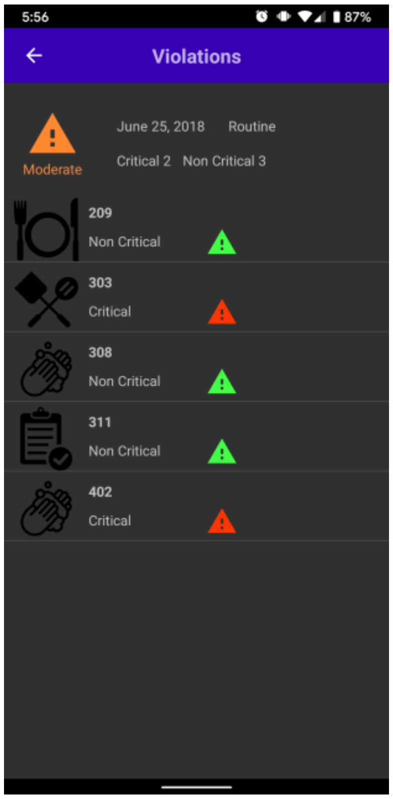
Favourite

Low
August 2019
Critical 1 Non Critical 0

Low
March 2019
Critical 0 Non Critical 2

Moderate
June 2018
Critical 2 Non Critical 3

Figure 2 (Screen 2)



5:56 87%

Violations

Moderate
June 25, 2018 Routine
Critical 2 Non Critical 3

209
Non Critical

303
Critical

308
Non Critical

311
Non Critical

402
Critical

Figure 3 (Screen 3)

Figure 1: A list of restaurants from the sample set.
Figure 2: The inspection reports in 104 Sushi & Co.
Figure 3: The violations for June 25 2018 inspection report in 104 Sushi & Co.

Iteration 2

Get updated data.

- Once the app is launched and it has been 20 hours since data last updated, check City of Surrey's server to see if there is a recent data to be downloaded.
- If there is new data on server, app should ask user if they want to update the data.
    - If user do not want data to be updated, app should ask again in next app start up.
    - App must only download data if a new update is available.
- Application should store downloaded data locally.
- Initially, the app should start with sample data set.
- Add a please-wait dialog, while data is downloading.
    - Animation to show application is working.
    - Add a way to cancel the download.
    - Cancelled download should use previously downloaded data.

Map

- At application launch, the default view should be map, centred on user's current location.
- Peg should be on the map to show locations of each restaurant with data.
- Map should be moveable and zoomable.
- Restaurant pegs show hazard level of most recent inspection for restaurant.
    - Each peg show hazard level using colour and icon.
- Pegs should cluster, if too many pegs in an area.
- Current GPS position should be shown in a dot.
    - As the user moves, the dot should update to new location and the map follows.
- User should be able to interact with pegs.
    - Clicking on peg should display a small pop-up of restaurant details.
    - Clicking on the small pop-up should go to restaurant's full information screen.
- A clear way to toggle list and map screen.

Custom Images

- Some restaurants should have images.
    - At least 10 restaurants must have unique icons in list view.
    - At least 5 should have 4 or more locations.

Back Button Behaviour

- Toggle between map and list view is independent of back button behavior.
    - If user is in map or list, pressing back will exit the application.
- Pressing back from single restaurant should bring back to the either map or list view, depending on which user last visited.
- On single restaurant screen, tapping GPS cords should show the location on the map.

Iteration 3

Search/Filter

- Search on map screen, to limit markers.
- Search on list screen, to limit restaurants in list.
- Searching details:
    - Clear current search to see all restaurants.
    - Preserve current search, when switching map and list screens and when clicking on single restaurant and pressing back.

- When searching, only should see…
    - Whose name matches something the user types.
    - Whose most recent inspection has specific hazard level.
    - Which has had < = (or >=) N critical violations within last year
    - Which has been tagged favourite.
- Combine searches above, to find specific restaurants.

Favourite
- On restaurant detail screen, mark and unmark favourite restaurants.
- On list screen, favourite restaurants should stand out.
- When new data is downloaded, application should tell updates from favourite restaurants and show which restaurants has new data.

Internationalize
- App should support more than one language.
- When the app launches, the language should be the Android default language.
    - If application does not support language, default to English.
- All strings should be translated, except for
    - Data read out of data files.
    - Networking errors
- Translate one-line summaries of violations in violations list.
- Include language in readme3.txt.

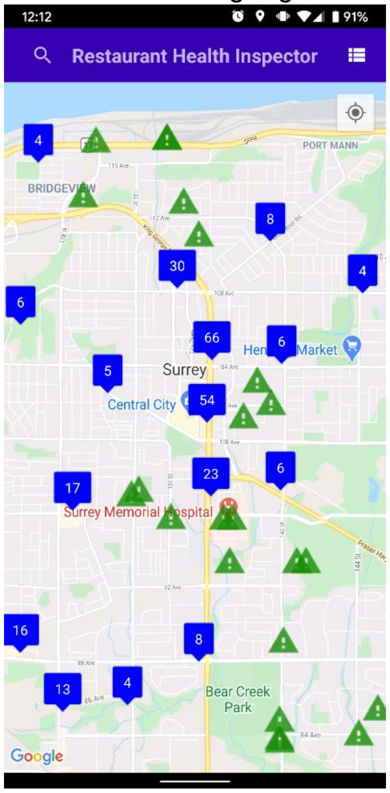
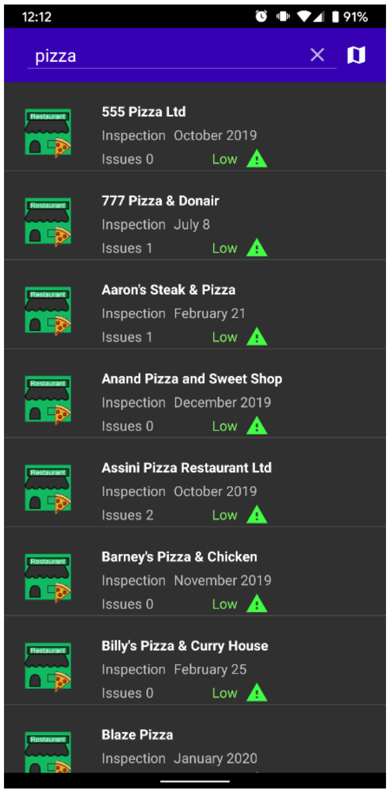


Figure 4 (Map view)

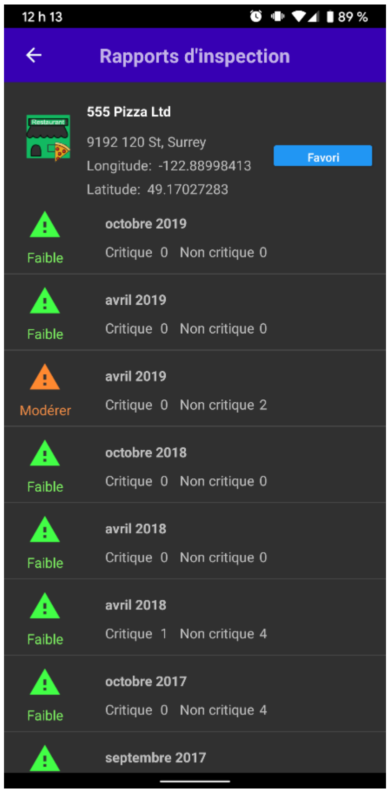


12:12 91%

pizza

555 Pizza Ltd	Inspection October 2019	Issues 0 Low
777 Pizza & Donair	Inspection July 8	Issues 1 Low
Aaron's Steak & Pizza	Inspection February 21	Issues 1 Low
Anand Pizza and Sweet Shop	Inspection December 2019	Issues 0 Low
Assini Pizza Restaurant Ltd	Inspection October 2019	Issues 2 Low
Barney's Pizza & Chicken	Inspection November 2019	Issues 0 Low
Billy's Pizza & Curry House	Inspection February 25	Issues 0 Low
Blaze Pizza	Inspection January 2020	

Figure 5 (Search)



12 h 13 89 %

Rapports d'inspection

555 Pizza Ltd
9192 120 St, Surrey
Longitude: -122.88998413
Latitude: 49.17027283
Favori

Faible	octobre 2019	Critique 0 Non critique 0
Faible	avril 2019	Critique 0 Non critique 0
Modérer	avril 2019	Critique 0 Non critique 2
Faible	octobre 2018	Critique 0 Non critique 0
Faible	avril 2018	Critique 0 Non critique 0
Faible	avril 2018	Critique 1 Non critique 4
Faible	octobre 2017	Critique 0 Non critique 4
	septembre 2017	

Figure 6 (Internationalize)

Figure 4: Map view of surrey with pegs and clusters.
Figure 5: Searching for pizza on list view.
Figure 6: Viewing the restaurant details of 555 Pizza Ltd in French.

3. Design

3a Architecture Overview

An Android Emulator or an Android phone is required to run Restaurant Health Inspection App. The Android must be at least Android OS 7.0 (API 24 Nougat) or newer. At least 50 MB of storage is required and the phone should have at least 1 GB of RAM.

4. Installation

4a Installing from Source Code

Follow these steps below to install the Restaurant Health Inspection App:

1. Download and open Android Studios on a computer.
2. Go to GitLab's website and search for "Restaurant Health Inspection App".
3. Once the repository is found, copy the clone link.
4. On Android Studios, click on File, New, and Project from Version Control.
5. Paste the copied link into URL, with Version Control set as Git.
6. Enter the login credentials to allow access to the GitLab project.
7. Wait for the project to be loaded into Android Studios.
8. Install Android Emulator or plug in an Android Phone via USB.
9. Press the Run button to download and start application.

5. Project Management

5a Software Development

Programming Languages Used

Java is the only programming language used for the application.

Development Platform

Android Studios is the development platform and can be run on Windows, Mac, Linux and Chrome OS. 4 GB of storage and 8 GB of RAM is recommended to use Android Studios. An Android Emulator or Android phone must be used to test the application.

Version Control System

GitLab is used as the version control system. Issues with branches will be used to develop each feature. The branches will be merged to master, after the features is fully developed with no bugs. Issues will be managed by the Product Owner and merge requests will be handled by the Repo Manager. (More information about the roles are in the next section)

5b Process Organization

Scrum

Scrum is a type of agile process, where sprints and roles are used. Below has more explanation on Scrum Sprints and Scrum roles.

Scrum Sprints

Each Scrum sprint (iteration) is two weeks long and there are three iterations. At the start of each sprint, new user stories will be created. The team will work on the software and deliver a working Android app at the end. Retrospective will be done, giving peer feedback to other team members.

Scrum Roles

Scrum roles are given at the start of the sprint and will be changed in each iteration. Below are the list of Scrum roles and their responsibilities.

Scrum Master
- Plans and organizes meetings, making sure they happen.
- Ensures that the team communicates and works together to solve problems.

Product Owner
- Askes the customer for project clarification.
- Creates issues in GitLab for each feature.

Repo Manager
- Setups Git and GitLab.
- Accepts merge request and help code reviews.

Team Member
- Codes more than the other roles, with no additional duties

5c Milestones & Schedule

Milestones	
June 24, 2020	Groups made and Start Iteration 1
July 9, 2020	End Iteration 1 and Start Iteration 2
July 22, 2020	End Iteration 2 and Start Iteration 3
August 5, 2020	End Iteration 3

Table 1: Project milestones from the course CMPT 276.

5d Management

Risk Management

Strengths

All team members have previous experience in Android Studio and Java. Teammates understand the methodology behind Scrums from lectures.

Weakness

Teams are chosen randomly and no previous working experience. With Covid-19, meetings are difficult to plan, with some teammates living in different time zones.

Resources

[1] “Download Android Studio and SDK tools.” Android Developer. https://developer.android.com/studio (accessed October 29, 2020)

[2] F. Jeff. “Android Studio for beginners, Part 1: Installation and setup.” Infoworld. https://www.infoworld.com/article/3095406/android-studio-for-beginners-part-1-installation-and-setup.html (accessed October 29, 2020).

[3] “SFU CS Instructional Gitlab.” GitLab. https://csil-git1.cs.surrey.sfu.ca/ (accessed October 29, 2020)

[4] T. Jack. “Project.” CMPT 276 – D200 – Fall 2020. https://opencoursehub.cs.sfu.ca/jackt/grav-cms/cmpt276-2/projects (accessed October 29, 2020).