System Requirements Specification

for

Earthquake Risk Assessment

Version 2

PREPARED FOR
COMPSCI 2XB3

PREPARED BY
Group: 01
Kan Hailan, 400207974
Sembakutti Kalindu, 1046206
Tao Haoyang, 400171589
Ye Fang, 400273067

March 23, 2020

Revision History

Name	Date	Version
Earthquake Risk assessment	7 March, 2020	1
Earthquake Risk assessment	22 March, 2020	2

1 Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the java application for earthquake risk assessment. It will explain the purpose and features of the application and what the application will do. This document is intended for the developers, testers, project leader and users of the application.

1.2 Domain

Our application is developed for individuals and insurance companies. Both of them have access to search and risk assessment functionalities. Users are allowed to enter a location and our app will not only list past earthquake information but also give a risk assessment based on earthquake frequency, magnitude and population density. Individuals can search for historical occurrences of earthquakes near a specific location and consider the risk of the place they are settled in. Insurance companies can use the earthquake history as reference data and take the risk as a factor when they are making earthquake related insurance plans for houses, vehicles and people to ensure that they are making profit. Since individuals and insurance companies are sharing the same datasets, individuals can check our app to make sure that prices are reasonable when they are buying insurance, which raises their confidence towards the company. In the meanwhile, the company's reputation and credibility are largely improved among customers. Our app is devoted to achieving a win-win situation between insurance companies and individuals.

1.3 Definitions, acronyms, and abbreviations

ERA - Earthquake Risk Assessment

1.4 References

830-1998 - IEEE Recommended Practice for Software Requirements Specifications

https://www.docsity.com/en/ieeexplore-srs-template/591509/

Earthquakes in Canada

https://open.canada.ca/data/en/dataset/4cedd37e-0023-41fe-8eff-bea45385e469

Population and dwelling counts, plus data for subdivisions (municipalities)

https://open.canada.ca/data/en/dataset/402495f0-6415-4fd8-bcfc-25a304e6fc8b

List of Cities with coordinates in Canada

https://www.latlong.net/category/cities-40-15-4.html#

2 Overall description

2.1 Product perspective

ERA is developed for people who care about the potential earthquake risks in their target areas. They can search the historical earthquake information for a specific location and get the relative earthquake risk for the location. The app is developed to run in a java environment. We will have a module to inject advertisements for insurance companies to provide users with multiple earthquake insurance plans.

2.2 Product functions

Search: search the earthquakes for a given location and radius in the dataset
Sort: sort a list of earthquakes based on magnitude or their distance to the given location
Risk Assessment: calculate the relative risk rating base on the historical earthquake frequency, average magnitude, population density in the cities within 100km from the given location
Display the search results: display the list of earthquake information in different order
Display the assessment results: show the relative risk rating base on the risk assessment formula; display the historical earthquake frequency, average magnitude and population density in the assessing area; recommend a nearest city with a lower risk rating

2.3 User characteristics

- Primary users: Insurance companies, who want to use ERA as a reference for making earthquake insurance plans.
- Individuals: Such as homeowners, who want to use ERA to assess the earthquake risk in their community and decide if it is worth buying earthquake insurance which is not covered by homeowners' insurance.

2.4 Constraints

Data constraints:

The app only shows the correct result when the users enter a location within Canada based on the datasets constraints.

2.5 Assumptions and dependencies

- The future earthquake risk in a specific location is related to the historical earthquake frequency, average magnitude as well as the current city population density within 100 km from the location. We calculate the relative risk rating based on the three parameters. The assuming risk rating for the three factors are shown in the following table. The overall risk rating is the sum of the risk rating of the three factors.
- Software is dependent on access to the internet. We will use the Google Mapkit interface to get the user's location and convert the location to the global position coordinates.

Table1. the assuming risk rating of frequency, average magnitude and population density

frequency	risk rating		average magnitude	risk rating		population density	risk rating
< 1	0		< 1	0		< 1000	0
1 - 10	1		1 - 4	1		1000 - 5000	1
10 -100	2		4 - 6	2		> 5000	2
100 -1000	3		6 - 7	3			
> 1000	4		> 7	4			

3 Specific requirements

3.1 External interface requirements

user interface:

- **Search:**
    - Location: chooses a location on the map and convert the location to global position coordinates
    - Radius: enters a number with a unit of km
    - DisplayByMagnitude: displays the list of earthquakes in descending order based on their magnitude
    - DisplayByDistance: displays the list of earthquakes in ascending order based on their distance from the query location
- **Risk Assessment:**
    - ShowRisk: displays the relative risk assessment result; displays the earthquake frequency, magnitude and population density for the given location; displays the nearest city with lower earthquake risk rating

3.2 Functional requirements

Product Backlog:

Our application produces a list of earthquakes that had happened within close proximity for a specific location. The application also generates an earthquake risk rating for a specific location, based on the frequency, magnitude and population density.

ID	Description	Completion
1	Create a CSVreader to read earthquakes, populations, and city coordinates	
2	Create initial use relationships of application modules	

3	Display earthquakes within a given radius of an input location	
4	Set up EarthquakeT to accurately represent earthquake data	
5	Set up PointT, CityT, CityPostT to accurately represent geographical location data	
Sprint 1	Transform data from .csv files to java objects	100%
6	Search EarthquakeT within a given radius of input location	
7	Sort EarthquakeT objects based on magnitude and distance using Insertion Sort & QuickSort	
8	Represent CityT objects using a Symbol Table	
9	Represent CityPostT objects using a directed weighted graph	
10	Calculate a risk rating using both earthquake and population data	
11	Calculate and find another city from the connected graph which has lower risk rating than a specific city	
Sprint 2	Completion of Internal Workings of the application	100%
12	Make an input form to take user inputs	
13	Make a scrollable list to display earthquakes for an input location	
14	Give choice to display a list of earthquakes sorted in different orders	
Sprint 3	Application with an interactive user interface	80%
Release 1	Public version without advertisements	
15	Inject google advertisements using a module	
16	Change current location automatically according to location detection	
Sprint 4	Location detection & advertisements display	
Release 2	Second version with Advertisements from stakeholders	

3.3 Nonfunctional requirements

Human-computer interface:

- Our application displays a searchable list of earthquakes in a java environment
- Our application assigns a color-rating based on the magnitude of the earthquake

Reliability:

- We expect the Graphical User Interface to be separate from our application logic, in order to make GUI easier to update, and also to maintain reliability across future updates to the GUI.
- The application adheres to functional requirements when implementing functionality of the modules.
- We will provide a clear scope of our software service, and make assumptions known to the user.

Accuracy of results:

- We expect the generated list of earthquakes for a specific location to be 100% accurate.
- We expect the average population for a specific location to be 60% accurate.

Performance:

- We expect to display a list of earthquakes for a specific location within 30 milliseconds of user submission of an input location.
- We expect to generate a risk rating for a specific location within 1 second.

Software quality attributes:

- We will track the modules using Module Interface Specification to ensure accurate naming and implementations of functionality of modules.
- We will use modules to encapsulate related functionality of our application.

3.4 Requirements on the development and maintenance process

- Quality Control: After implementing a function, the individual will have to make sure his or her implementation works by doing a simple unit test. And when all the pieces are put together, some of the team members will be in charge of doing integration tests. Finally, do system tests by:
    - 1.make test plans
    - 2.make up test cases
    - 3.implement test classes
    - 4.locating bugs
    - 5.bug fixing
    - 6.move to step 1.
- Priorities of the required functions: The Earthquake ADT and City ADT and related functions (set function, get function .etc) have the highest priority to implement. And then, algorithms related functions. Finally, other functions.
- Likely changes to the system maintenance procedure: We might add a phase to provide new features of our program, and a phase to locate potential bugs.
- Other requirements
    - Our program needs to be correct, but not necessarily robust.
    - Our program needs to be reliable which means it does what is intended to do.
    - Verifiability is important.
    - And good reusability of code.