Software Requirements Specifications

Bug Similarity in Heterogeneous Networks (Research Project)

Prepared by:

Group Name: **CluB_Elite**

Harsh Goyal	**IIT2018114**	**B.tech IT Sec-B**
Aaditya Gadhave	**IIT2018144**	**B.tech IT Sec-B**
Sourabh Gupta	**IIT2018149**	**B.tech IT Sec-B**
Meet Singh Gambhir	**IIT2018158**	**B.tech IT Sec-B**
Tushar Atrey	**IIT2018159**	**B.tech IT Sec-B**

Instructor:	**Mr.Amit Kumar, Mr.Ashutosh Kumar**
Course:	**Software Engineering**

Date: **06/02/2020**

TABLE OF CONTENTS

1 Introduction

With the help of the connections between any two bugs , .i.e. all the types of relationships possible , in relating to the bugs using this we need to get out to what extent two of any bugs are similar. Based on that , if the two are changing one type of files , working on the same OS , are upto same component and also on the values of such similarity paths.

1.1 Purpose Of The Document

Document's Purpose -> We are considering that if the two bugs work on same type of files , then there are more sort of similarities between them , than if they do not do so. And similar can be said for the other types of relations as well. And assuming these factors , if true for two of bugs , their similarities should come out to be more than the two , which lacks this factor. So as to use this information in the near future , to predict something and use it to our advantage.

1.2 Product Scope

Heterogeneous graph is one of the latest concepts which has gained a lot of attention in recent times as due to the advantage of its over that of the homogenous one as it contributes very more of that in retrieving the information than the homogenous one.
The main reason for this being that it considers all types of possible elements of the network as nodes , and any edge between the two , ensures their connectivity , whether they being of different types of nodes , which is not being the case when we deal with the homogenous one.
It will use that there will be some sort of similarities between two nodes based on the heterogeneous graph .It has various useful applications such as predicting relationships between various possibilities of two nodes possible and not only the two bugs only. Somewhat similar concept is also used in the link prediction and in the social media suggestion providing.

1.3 Definitions, Acronyms and Abbreviations

Acronyms and abbreviations:

a. *SEOSS dataset*
b. *SRS: Software Requirement Specification*

Definitions:

a. *SEOSS Dataset:* A systematically retrieved dataset consisting of 33 open-source software projects containing a large number of typed artifacts and trace links between them. The artifacts stem from the projects' issue tracking system and source version control system to enable their joint analysis.
b. *Heterogeneous Graph:* A heterogeneous graph is a type of graph that is comprising of various types of nodes (all nodes , i.e. possibility of all elements in the project, is to of the node) and edges (edges can be all , relating two nodes whether of same type or different types of nodes).

1.4 Acknowledgments And References

a. *IEEE SRS Format*
b. *SEOSS dataset*

2 Overall Description

2.1 Product Functionality

Our product is designed for finding the similarities between different bugs , understanding different kinds of relationships between bugs,different kind of meta paths between the bugs, extent of relationship between bugs,their research papers and many other purposes of project.This would help in finding the duplication in bugs ,whom to assign a particular bug and to group the similar kind of bugs.

2.2 Operating Environment

We are taking a shot at enormous information heterogeneous systems. We are given information from 33 distinct projects as sqlite datasets. Sqlite is a piece of python's standard library . We are figuring individual relationship with the assistance of google sheets. We are submitting and pushing the progressions to our python application at github.

2.3 User Characteristics

Clients ought to be comfortable with running projects utilizing command prompt. He/She ought to be comfortable with different criteria of similarity, similar to individual coefficient and path based importance measure.

2.4 **Product Functions**

Use Cases	**Description of use case**
Admin:	
Login	Allows admin/user to login.
View Data	Allows admin/user to view the SQL database to our application
Make Heterogeneous Graph.	From given database Admin creates the required Heterogeneous Graph
Number of the paths for metapath.	Calculate total number of paths with the given type of metapaths+
Path based Relevance Measure	Finding the relatedness of an object pair depending on the given relevance path.
Testing data.	A dataset is added to our application for testing our algorithm and checking its correctness.
Design similarity matrix	Similarity Matrix is created by the training data which will further help in generating links.
Evaluation	Calculates accuracy between predicted links using different projects
Correlation different versions of the project.	Prediction of similarity between bugs in different versions of project.
User:	
Login	Allows admin/user to login.
Insert Data	Allows users to insert data about the bugs of the project.

View Data	Allows admin/user to view the SQL database to our applicationUser can also view the whole similarity matrix
Modify Data.	Users can modify the existing database.
Choose meta paths.	Admin gives different types of meta paths to find the links between Bugs.
Result	From given database Admin concludes outcome from Heterogeneous Graph
Duplicate Bugs.	From the result user can view all the duplicate bugs of the data.
Similar Bugs.	From the result the user can view all the Similar bugs of the data.
Invalid Bugs.	From the result the user can view all the Invalid bugs of the data.
Single Bug Status	From the result the user can view the status of any particular bug.
Bug Categorisation.	From the result the user can view all the classification of the bugs of the data.

3 Specific Requirements

Functional Requirements

We describe the functional requirements by giving various use cases:

Use Case 1:

Name: Login

Summary: Allows admin/user to login

Actors: Admin/User

Main success scenario:
• Admin/User clicks on the login button.
• App checks for the Verification of login.

Extension: Id or password incorrect. Shows error dialog box.

Post-condition: Admin/User can now access all features of the app.

Use Case 2:

Name: Insert Data

Summary: Allows users to insert data about the bugs of the project.

Actors: User

Main success scenario:
• Admin provides dataset in the form of sqlite database.
• App connects the sqlite database with the project and includes it for predicting Bug Similarities.

Extension: Here we are working with sqlite databases for this project, but we can also work with other databases after performing database migrations such as MySQL.

Post-condition: Project will now use the given database for predicting Bug Similarities.

Use Case 3:

Name: View Data

Summary : Allows admin/user to view the SQL database to our application

Actors : Admin/User

Main success scenario:
- Actor connects to the database successfully and verifies the data.

Post-condition: Admin can now access the data from different projects in our heterogeneous graph.

Use Case 4:

Name: Modify Data.

Summary: User can modify the existing database.

Actors: User

Main Success Scenario:
- Users cross-verifies the database and make required changes.

Post Condition: Now User can expect Correct Result.

Use Case 5:

Name: Make Heterogeneous Graph.

Summary: From given database Admin creates the required Heterogeneous Graph

Actors: Admin

Main Success Scenario:
- Successful formation and display of the required graph.

Extension: Now the link between the nodes can be viewed in the network.

Post Condition: Now we can Predict of similarity between bugs by using this graph.

Use Case 6:

Name: Number of the paths for metapath.

Summary: Calculate total number of paths with the given type of metapaths

Actors: Admin

Main Success Scenario:
- There are various types of meta paths in the dataset and calculation of total number of paths of the specified metapath between bugs.

Post Condition: Prediction of similarity between bugs by counting total number of paths of all the meta paths.

Use Case 7:

Name: Choose meta paths.

Summary: Admin gives different types of meta paths to find the links between Bugs.
Actors: User

Main success scenario:
- Admin provides various metapaths which will be studied as a criteria for prediction of Bug similarity.
- App measures the number of various metapaths to use path based relevance measure algorithm.

Post-condition: Project will now measure all the various kinds of metapaths, so that path based relevance measure algorithm can be applied.

Use Case 8:

Name: Result.

Summary: From given database Admin concludes outcome from Heterogeneous Graph

Actors: User

Post Condition: Now the User can see the details of the result.

Use Case 9:

Name: Duplicate Bugs.

Summary: From the result user can view all the duplicate bugs of the data.

Actors: User

Main success scenario:
- App measures all the Duplicate Bugs and shows it to the user.

Post-condition: All the Duplicate Bugs can be discarded.

Use Case 10:

Name: Similar Bugs.

Summary: From the result the user can view all the Similar bugs of the data.

Actors: User

Main success scenario:
- App measures all the Similar Bugs and shows it to the user.

Post-condition: All the Similar Bugs can be categorised into one category..

Use Case 11:

Name: Invalid Bugs.

Summary: From the result the user can view all the Invalid bugs of the data.

Actors: User

Main success scenario:
- App measures all the Invalid Bugs and shows it to the user.

Post-condition: All the Invalid Bugs will be discarded and not considered into the data.

Use Case 12:

Name: Single Bug Status

Summary: From the result the user can view the status of any particular bug.

Actors: User

Main success scenario:
- App measures all the Status of the particular chosen bug and shows it to the user.

Extension:
A particular bug can be fixed , or in progress , is invalid or can be among the duplicate ones or even can be an enhancement.

Use Case 13:

Name: Bug Categorisation.

Summary: From the result the user can view all the classification of the bugs of the data.

Actors: User

Main success scenario:
- App measures all the classification of the Bugs and shows it to the user.

Extension:
Different algorithms such as LCC membership , SVM classifier , LDA-Ki , Thresholding on eigenvector are employed to categorise the bug.

Post-condition: All the similar Bugs will be categorised in the same category..

Use Case 14

Name: Path based Relevance Measure

Summary: Finding the relatedness of an object pair depending on the given relevance path.

Actors: Admin

Main success scenario:
- Admin calculates the relatedness of bugs connected in the heterogeneous graph using the Path based Relevance measure algorithm given in HeteSim: A General Framework for Relevance Measure in Heterogeneous Networks.

Post-condition: Project will provide correlation between users(Developers) in a range from 0 to 1, using path based relevance measure algorithm.

Use Case 15:

Name: Testing data.

Summary: A dataset is added to our application for testing of our algorithm and checking its correctness.

Actors: Admin

Use Case 16:

Name: Design similarity matrix

Summary: Similarity Matrix is created by the training data which will further help in generating links.

Actors: Admin

Main success scenario:
- The similarity matrix between different types of nodes is created and it helps in link prediction techniques using this matrix.

Post-condition: Now the application contains a similarity matrix which can be further used for link prediction in our heterogeneous graph .

Use Case 17 :

Name: Evaluation

Summary: Calculates accuracy between predicted links using different projects

Actors: Admin

Main success scenario:
•Based on previous predictions we can calculate our accuracy percent by calculating the pearson coefficient.

Post-condition: We can know the accuracy of predicting a new link.

Use Case 18:

Name: Correlation between different versions of a project.

Summary: Prediction of similarity between bugs in different versions of project.

Actors: Admin

Main success scenario:
• The project checks newer versions of any given project to check if the link predicted by the app does actually exist in the future.

Post-condition: This will help to research that up to what extent does heterogeneous graph predicting links works.

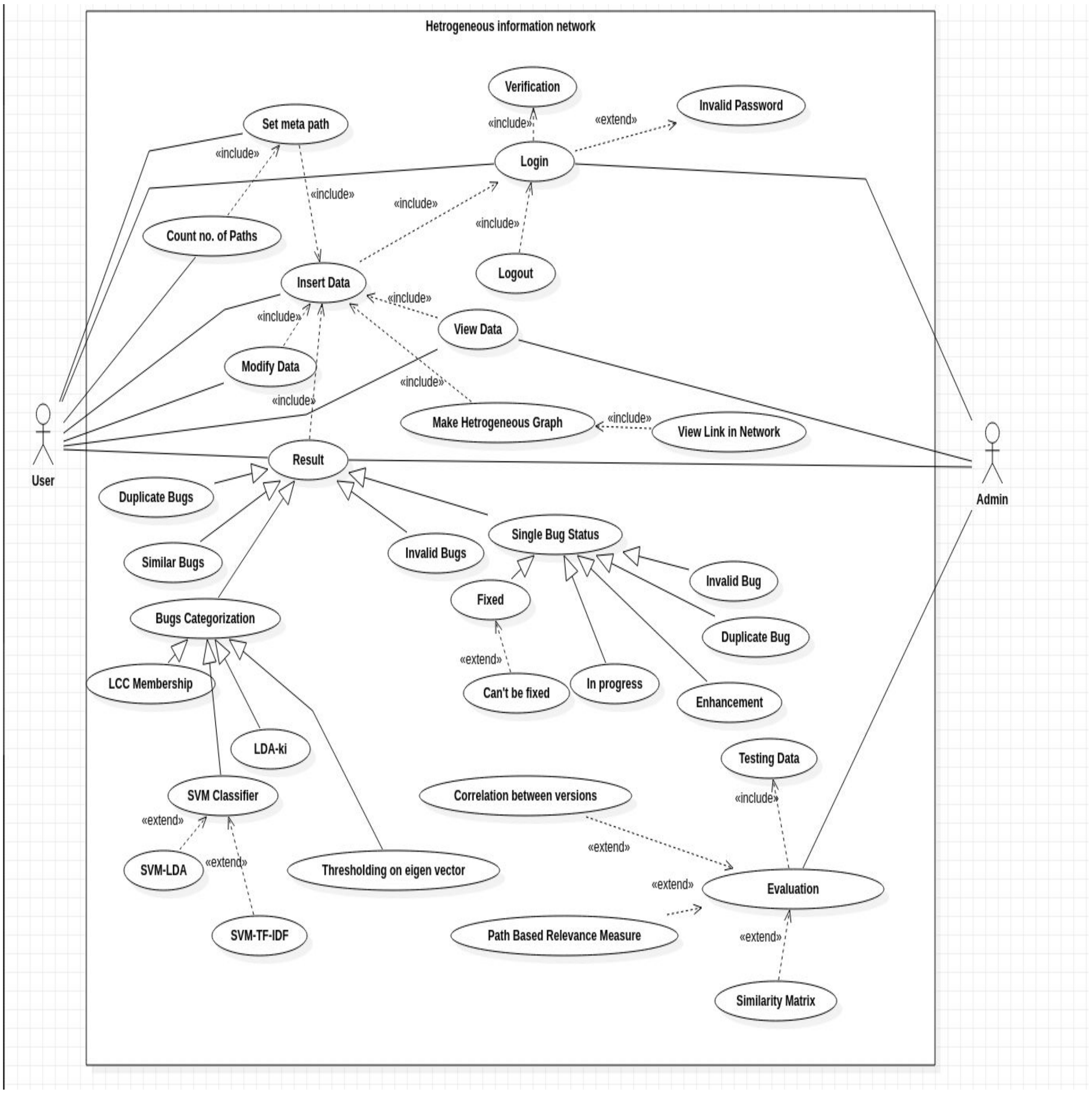
Main success scenario:
•.Training data is quite essential for generating appropriate results from the algorithm.
• A separate graph is created based on the training dataset in which we can test our algorithm to increase the accuracy of our algorithm.

Extension: The given dataset should be valid i.e it should contain all the required fields for our algorithm to work else an error message will be shown.

Post-condition: The algorithm is now trained and can be seen working correctly if not the necessary changes can be made so that it works fine on big datasets.

3.1 **Use Case Diagram**

3.2



4 Non-functional Requirements

Software Quality Attributes

Right now we are fundamentally making a heterogeneous diagram dependent on information of different engineers and bugs which they tackled already. We will foresee the idea of an engineer in the coming days. We will predict the relationship between any two bugs based on various factors such as the developer , the bug files that are changed , operating system,term of the bug, etc, and edges of the graph , relating the two nodes. All these factors will help us figure out the number of links between any given bugs and the possible number of links in future.

Since our applications utilizes all the calculations which have been as of now confirmed , along these lines results produced by our application would be very solid. The client information will be totally secure in our database and will be seen distinctly by client and administrator no other individual can get to the information.

5 Other Requirements

Aside from the calculation and database necessities of the task we additionally require precise preparing information which will prepare our venture as needs be. Preparing information should be sufficiently different to prepare our undertaking alright with the goal that it would itself be able to give exact outcomes. Since this is an AI based venture so exact preparing information is very basic .

We have additionally utilized a Sqlite Database for putting away our information and furthermore putting away our outcomes and likeness lattice. We likewise utilized a Sqlite GUI for working with databases and reviewing test information.

We additionally required different research papers to accumulate appropriate data about Bug Categorization and Meta Paths.

Appendix A - Group Log

1. Conversation about undertaking prerequisites, and research papers given to us.
2. Discussion “HeteSim: A General Framework for Relevance Measure in Heterogeneous Networks.”
3. Conversation over how to remember Heterogeneous charts for the venture, beginning from finding and perceiving Meta Paths.
4. Considering all types of meta paths between any given pair of bugs.
5. Connecting the sqlite database to the project, from the dataset given.
6. Considering all the future requirements of the projects by making Use Case Diagrams.