UNIVERSITY OF RIJEKA
FACULTY OF ENGINEERING
SEIP LABORATORY

Bug Finding in DGTool

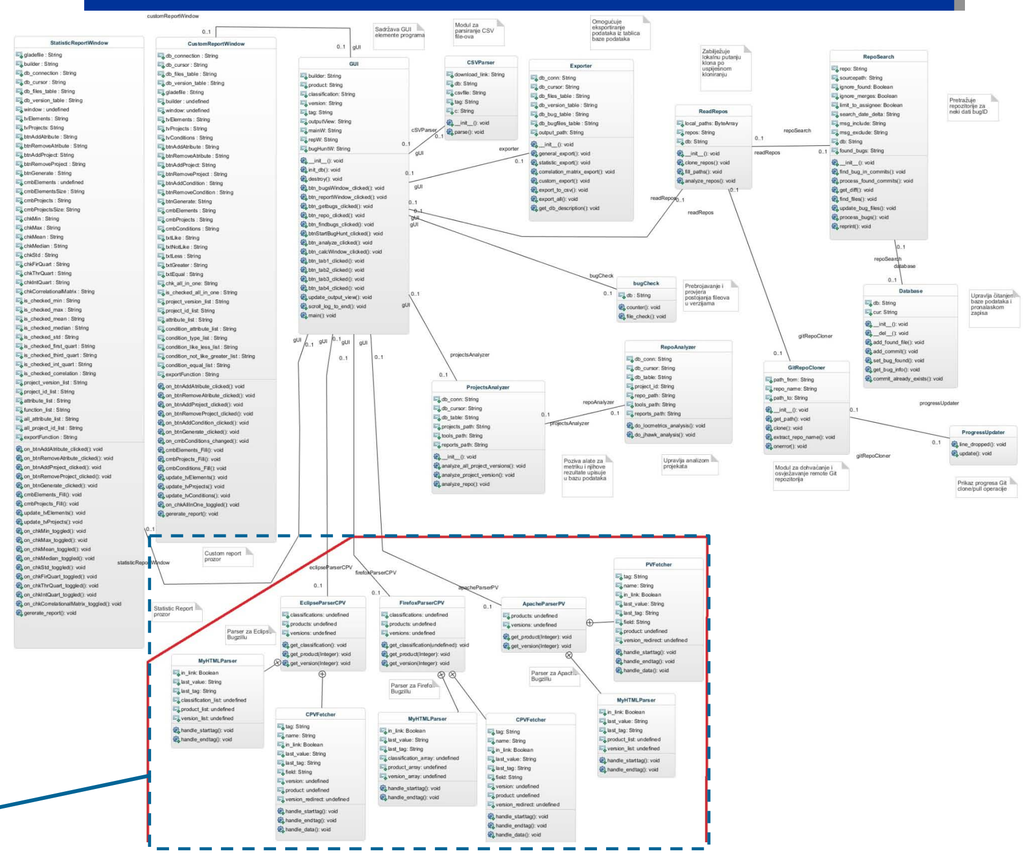
Mario Blažević, Nikola Domazet, Ozren Šejić

Summary

•Project goals:
- Fetch HTML data from Bugzilla
- Parse that data to a proper format
- Enable listing through data via GUI

• Libraries/tools required: PyGTK/Glade 3.6.1

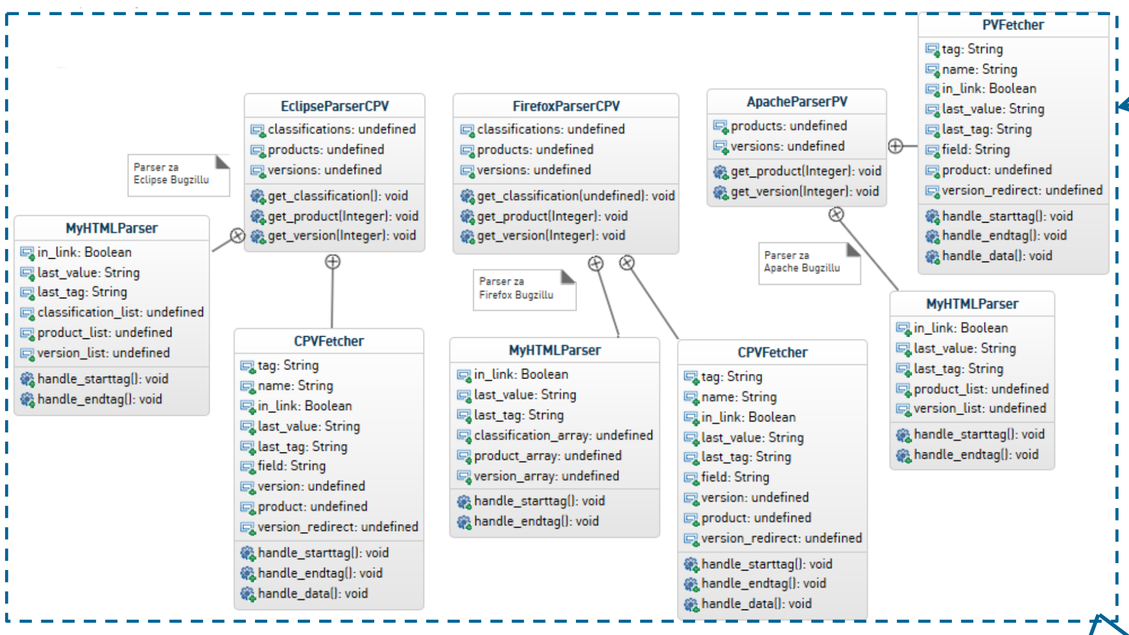
Overall Architecture



• **Architecture components:** Bug finding functionality, CSV parser, Project/ repository analyzer (using external tools), Repository maintainer, Repository history analyzer, Database abstraction utilities, reports building functionality

Developed Component

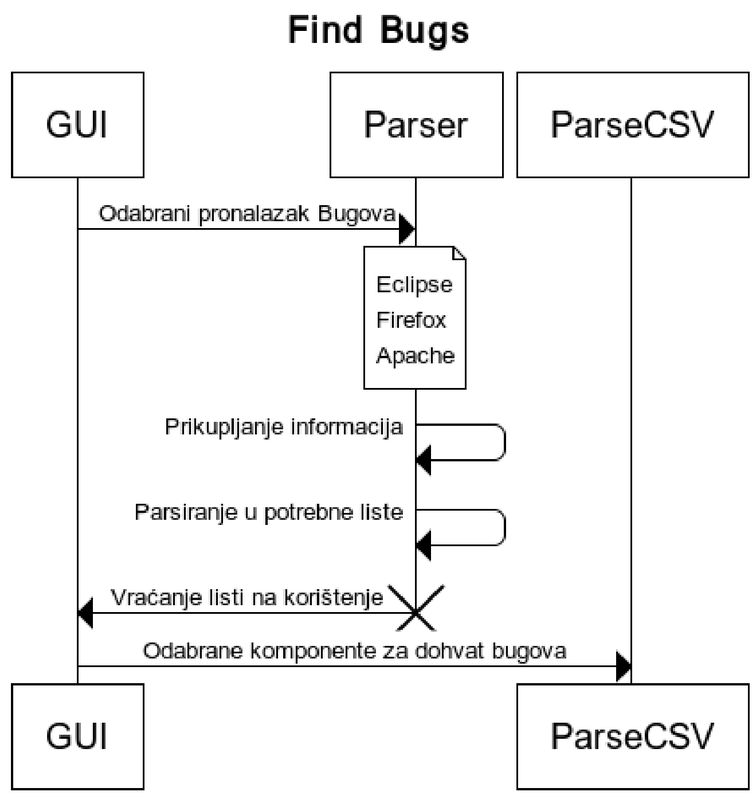
Class diagram



• There are three different parsers for bug finding operation on Bugzilla:
- Eclipse
- Firefox
- Apache

• Classes MyHTMLParsers and (C)PVFetchers are some sort of a inner classes for each type of chosen operation

Sequence diagram



• User can choose one of three parsers on GUI to use for finding bugs

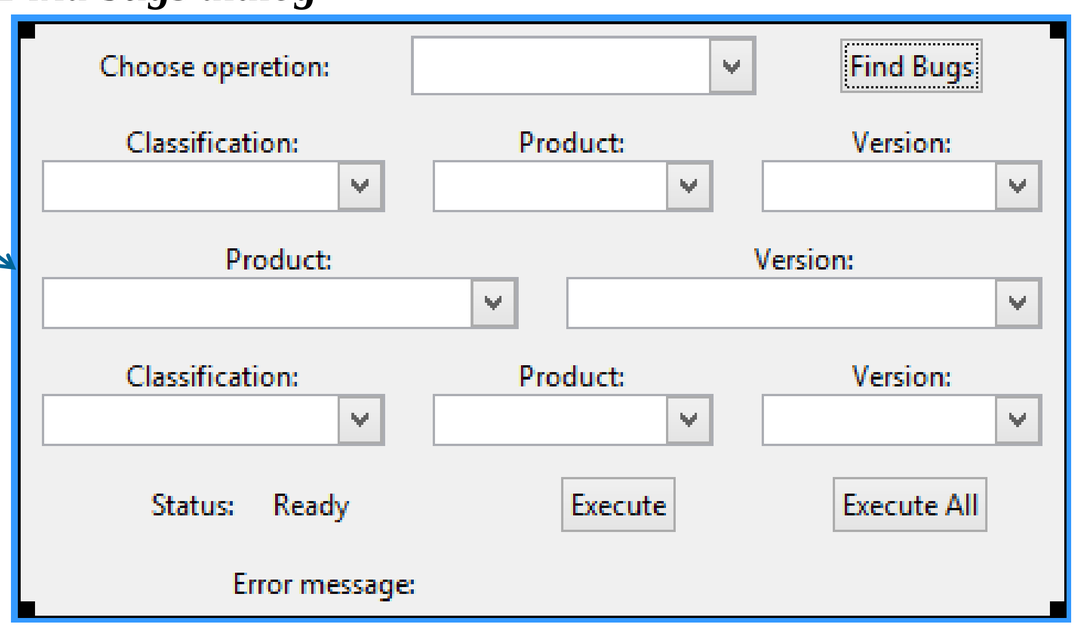
• The module then collects needed data and parses it in needed lists

• Those lists contain Classification, Product and Version which are again given to GUI component and user to choose each of them

• When choices are set user can initialize execution which transfers chosen data to ParseCSV module

GUI

Find bugs dialog



• The GUI presents the user with multiple choices of bug finding functionality

• First choice is "*Choose operation*" with executing button "*Find Bugs*"

• After process of finding is complete user can choose Classification, Product and Version

• With chosen parameters user can press "*Execute*" and "*Execute All*" to start fetching bug reports from Bugzilla and adding it to database through "*ParseCSV*" module

Conclusion

• Experienced problems:
  - Each Bugzilla part (Eclipse, Apache, Firefox) is different and demands single parser for each part

• Learned concepts:
  - Pyhon coding
  - OOP paradigm

• Future work improvements:
  - Enable multithreading
  - Proper documentation before development starts