

CS3203: Software Engineering Project

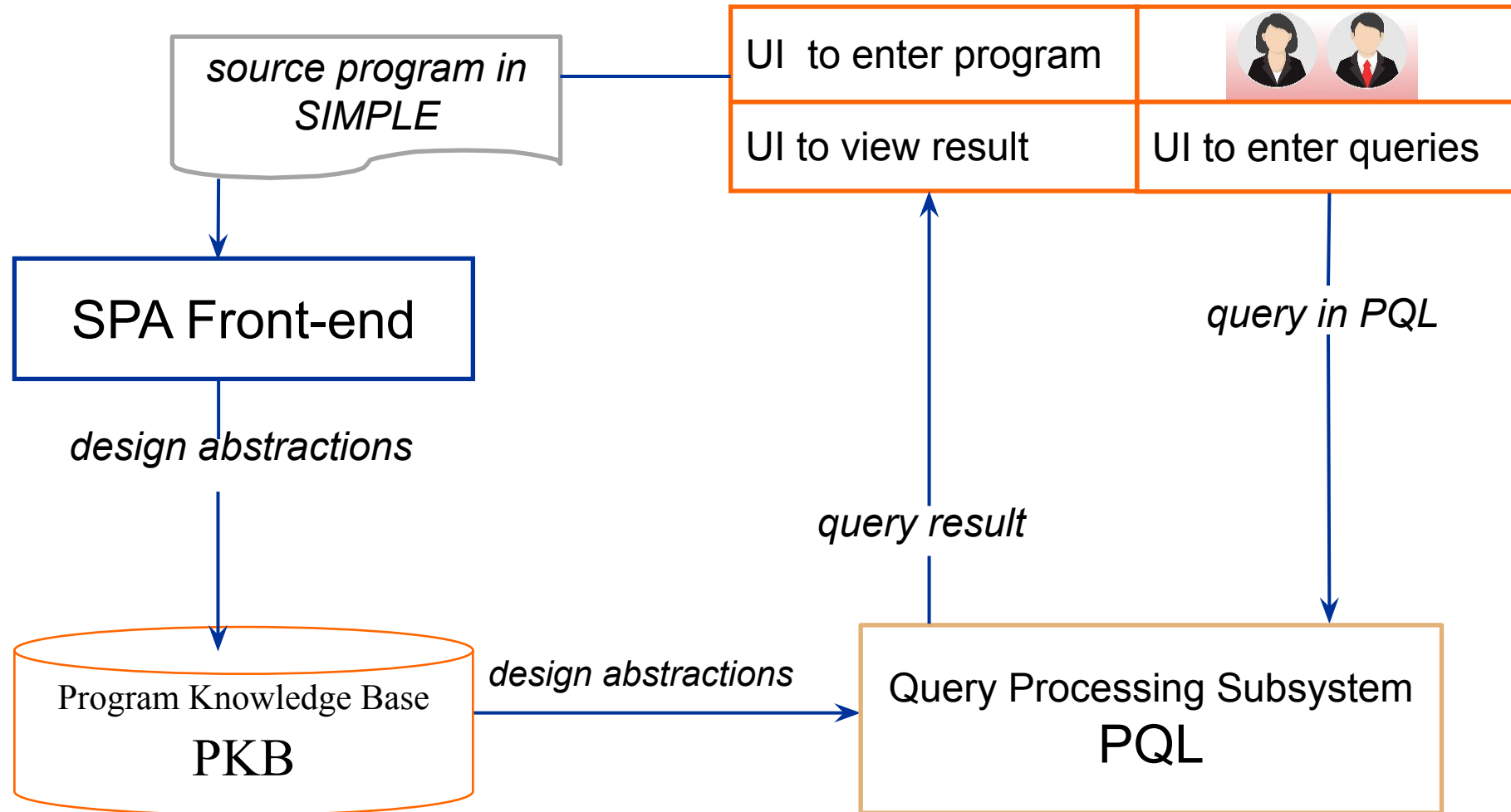
Tools for the Project

More info at - [CS3203 Wiki > Tools](#)



School of
Computing

SPA – Static Program Analysis Tool



Version Control System - Git

Make sure you brush up on your Git skills:

<https://try.github.io/>

Minimally, how to do the 3 things on Git

- **Git Add:** Add files to be committed
- **Git Commit:** Commit with message
- **Git Push:** Push to remote Git server



Git use in CS3203

- All teams **must** use the Git repository provided
 - Do not fork, use branches (Your tutors will be added)
- Each team must declare the info:
 - All GitHub IDs of the members of the team
 - Choice of startup solution (Windows or Cross-platform)
- Cross-platform, please also state the target environment (Windows, Mac or Linux)! See Wiki for more information!
 - **Make sure to test your system against the target env!**

Github Education - Git

The screenshot shows a web browser with two tabs: 'nus-cs3203/spa-19s1-win' and 'nus-cs3203/team15-win-spa-19s1'. The address bar shows 'github.com/nus-cs3203/team15-win-spa-19s1'. The GitHub navigation bar includes a search bar, 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The repository page for 'nus-cs3203 / team15-win-spa-19s1' is displayed, noting it is 'Private' and 'generated from nus-cs3203/spa-19s1-win'. It shows 4 watches, 1 star, and 0 forks. A secondary navigation bar includes 'Code', 'Issues 33', 'Pull requests 0', 'Actions', 'Projects 1', 'Wiki', 'Security', 'Insights', and 'Settings'. The repository description is 'No description, website, or topics provided.' with an 'Edit' button and a 'Manage topics' link. A progress bar shows 654 commits, 14 branches, 0 packages, 0 releases, and 7 contributors. Below this, a 'Branch: master' dropdown and a 'New pull request' button are visible, along with 'Create new file', 'Upload files', 'Find file', and a 'Clone or download' button. A commit history table is shown, with the latest commit by 'kengwoon' merging pull request #220 from 'nus-cs3203/Development' on Oct 22, 2019. The table lists files: 'Team15' (Add system test files, 3 months ago), '.gitattributes' (Initial commit, 5 months ago), '.gitignore' (Add gitignore, 4 months ago), and 'check-submission.py' (Initial commit, 5 months ago). At the bottom, a blue box prompts to 'Add a README with an overview of your project.' with an 'Add a README' button.

nus-cs3203 / team15-win-spa-19s1 Private
generated from nus-cs3203/spa-19s1-win

Watch 4 Star 1 Fork 0

Code Issues 33 Pull requests 0 Actions Projects 1 Wiki Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

654 commits 14 branches 0 packages 0 releases 7 contributors

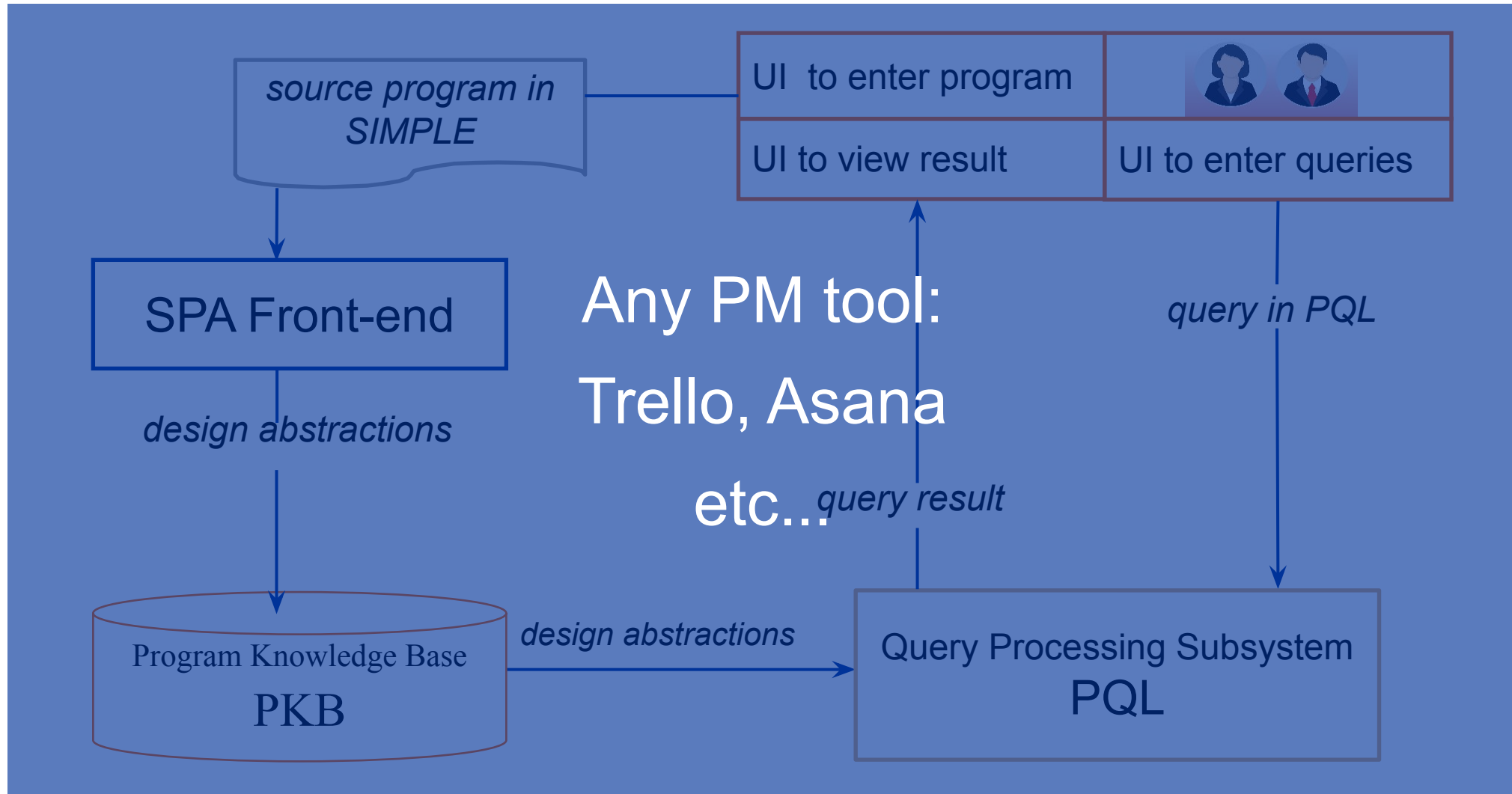
Branch: master New pull request Create new file Upload files Find file Clone or download

kengwoon Merge pull request #220 from nus-cs3203/Development Latest commit e71f632 on Oct 22, 2019

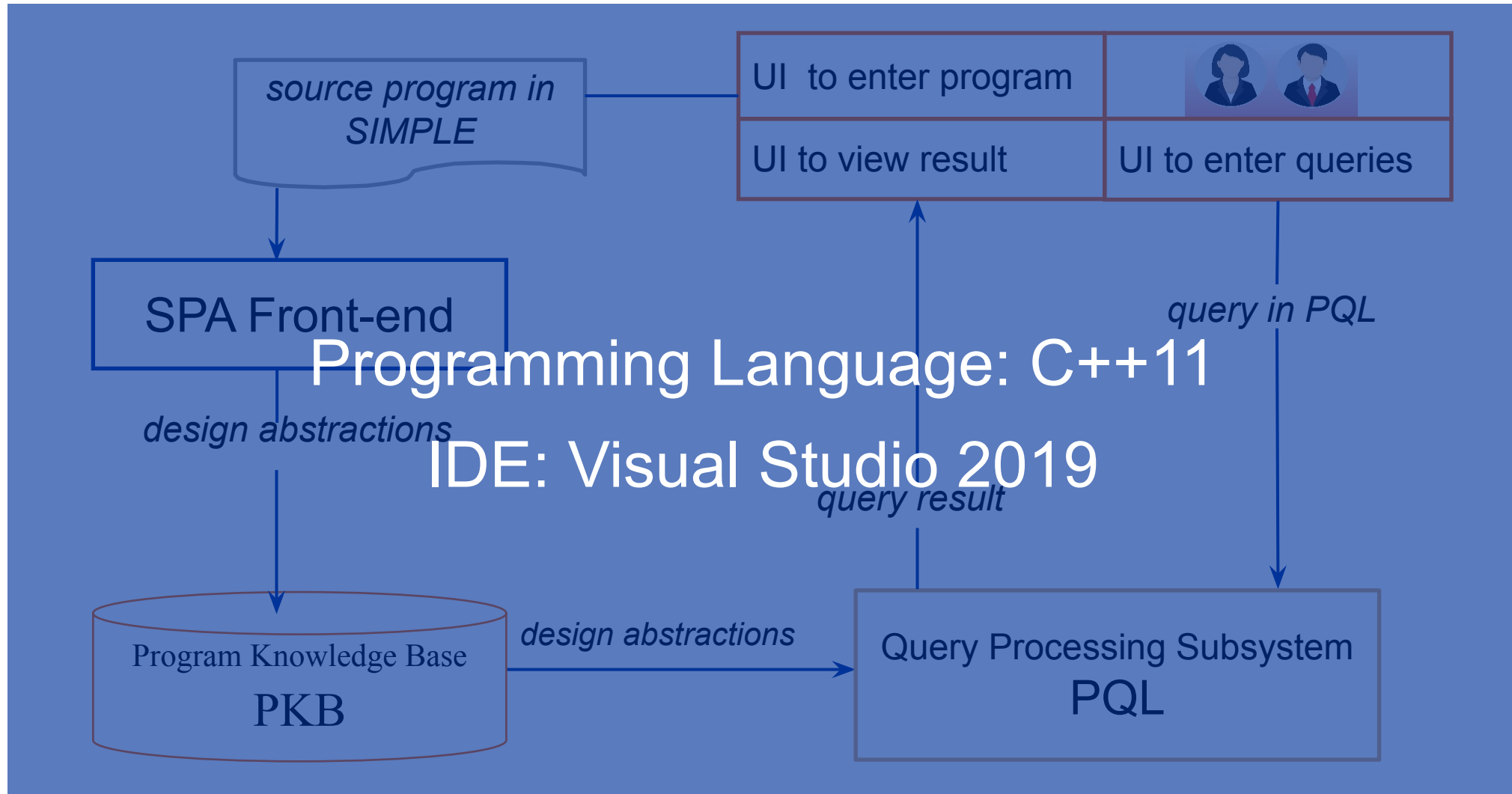
Team15	Add system test files	3 months ago
.gitattributes	Initial commit	5 months ago
.gitignore	Add gitignore	4 months ago
check-submission.py	Initial commit	5 months ago

Add a README with an overview of your project. Add a README

Project Management



Development Environment



Startup SPA Development

- Windows Startup SPA Solution (Official/Recommended):

- VS2019 Enterprise

Easy

- Cross-platform Startup SPA Solution:

- IDE:

- » VS2019 Enterprise

- » Clion with Make

- OS (See Target Environment):

- » Windows

- » MacOS

- » Linux

Not-so-easy

Notes on check-submission.py

Its a python tool that you should run to make sure your project satisfy the basic submission requirements (you should still check the requirements)!

```
$ python check-submission.py
```

This script will check for basic compliance with the submission requirements.

Disclaimer: you are still responsible for your submission, this check is by no means complete.

[Failed] - Team number must be valid.

Visual Studio

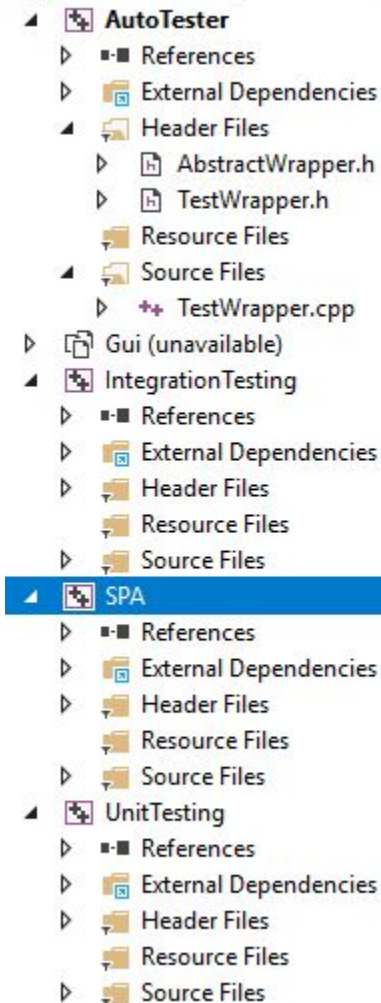
Windows

Open Startup SPA Solution

- The easy way to start!
- Available in "Tools" folder in IVLE Workbin
- Important files:
 - StartupSPASolution.sln
 - Read the documentation!
- Open and build!

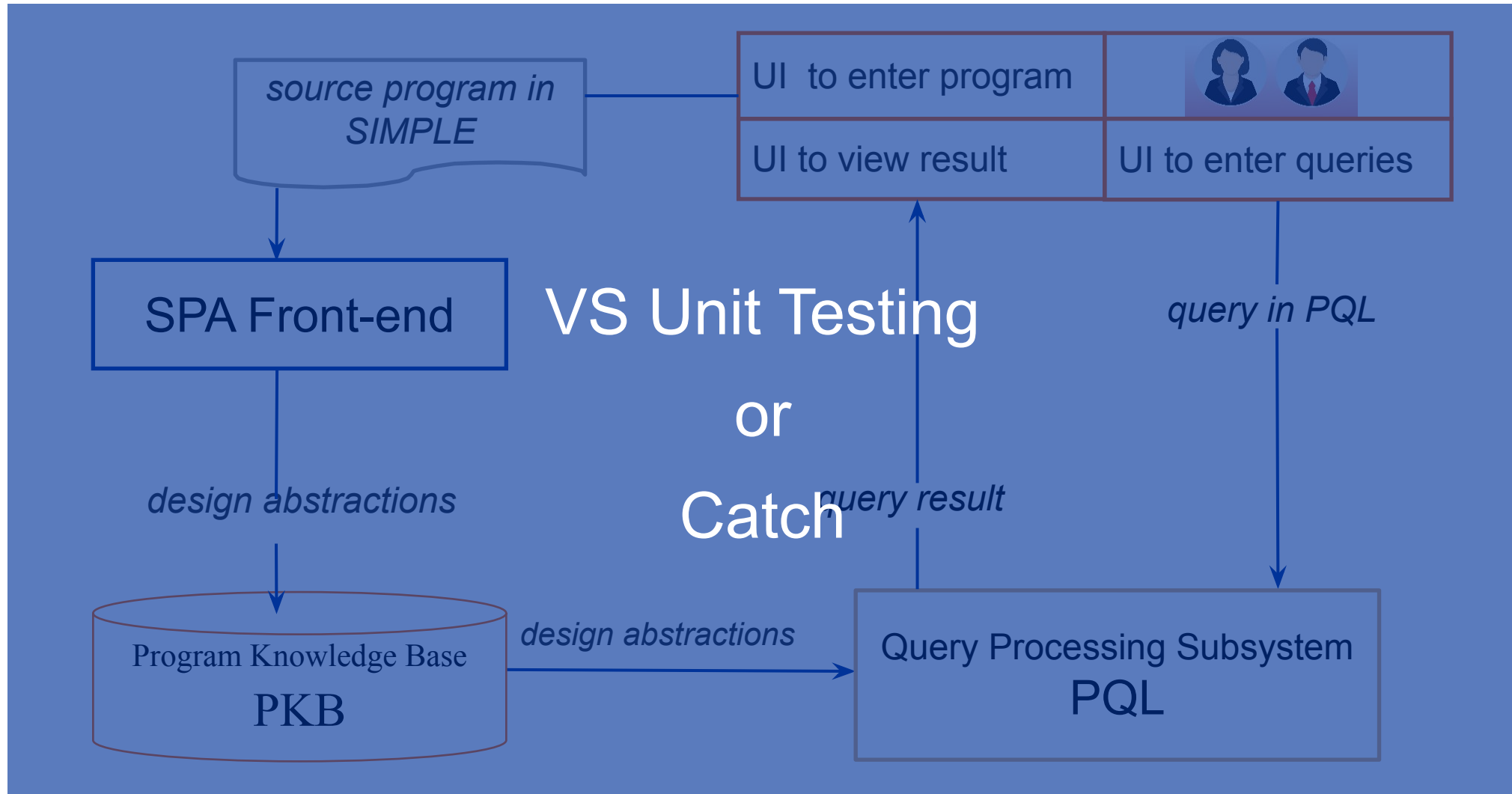
Analyze the Project Properties

Solution 'StartupSPASolution' (4 projects)



- **AutoTester:**
Build to get AutoTester.exe for your SPA
- **IntegrationTesting:**
Implement IntegrationTest.cpp and test classes, build and run for integration testing for your SPA
- **Empty SPA project:**
Fill in the code for your SPA
- **UnitTesting:**
Implement UnitTest.cpp and test classes, build and run for unit testing for your SPA

SPA Testing



Project Properties

- Compiling:
 - Turning source code into object code
 - VS2019: Additional Include Directories
- Linking:
 - Combining all the object code with the libraries into binaries
 - VS2019: Additional Dependencies
- Building:
 - The whole sequence from compiling to linking

Debug vs Release

- Debug:
 - PDB files are created: think of a lot more code added into your code to enable debugging
 - Not optimized as much
 - Autotester settings
- Release:
 - Optimized
 - What we will run our tests on

Running the AutoTester

- Running from the command line (cmd in Windows)
 - > AutoTester.exe ..\Tests\Sample_source.txt
 ..\Tests\Sample_queries.txt
 ..\Tests\out.xml
 - Open output.xml in Mozilla Firefox to see the results
 - Note: analysis.xsl is in the same directory with output.xml
- Running from VS in Debug mode
 - Press the green Run button
 - Open out.xml in Mozilla Firefox to see the results

GUI vs Autotester

- GUI:
 - Optional
 - For your own benefit
- Autotester
 - Implement your code into autotester
 - Mandatory
 - Will be used in grading

CMake

Windows, Mac, Linux

Cross Platform... But...

We strongly recommend you to use the same OS/IDE/Build system across every member!

- Windows / VS / CMake
- MacOS / AppleClang / CMake
- Linux(Fedora 30) / GCC / CMake

You can mix OS/IDE/Build Systems... But...

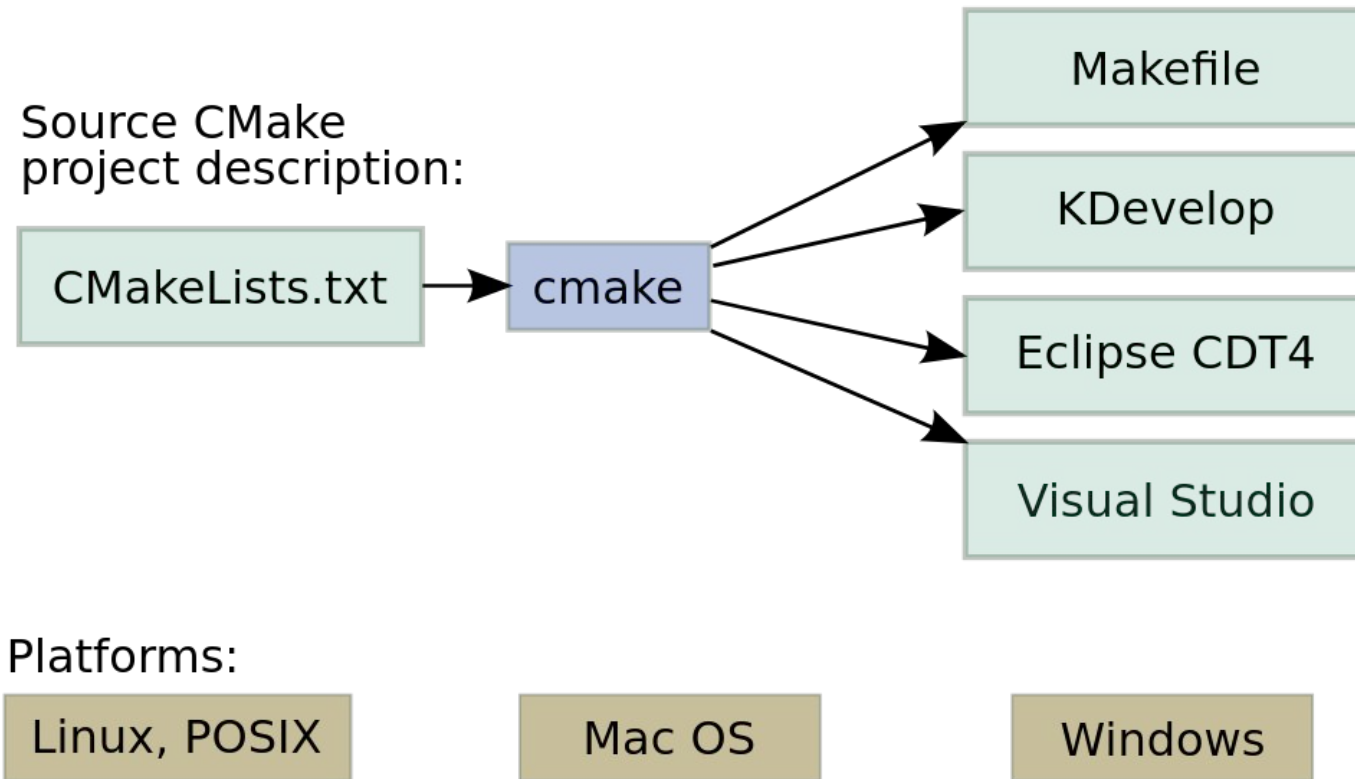
No additional marks will be given.

Support will be given on best-effort basis.

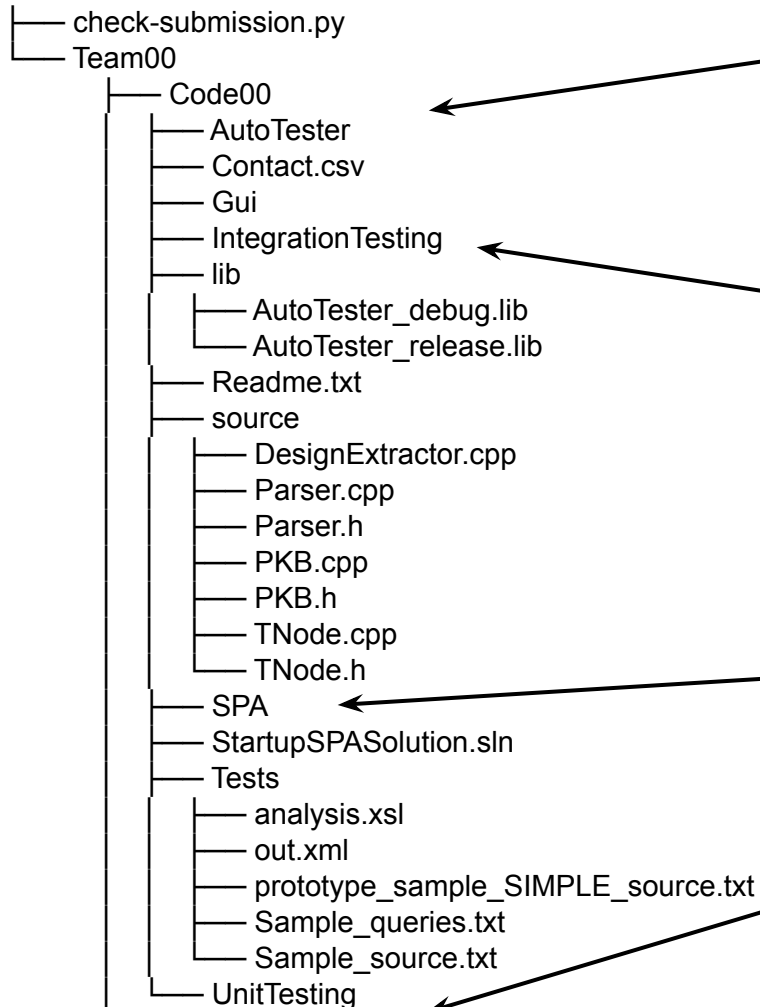
CMake

Make sure you check out the tutorial:

<https://cmake.org/cmake/help/latest/guide/tutorial/index.html>



Analyze the Project Properties



- **AutoTester:**
Build to get AutoTester.exe for your SPA
- **IntegrationTesting:**
Implement IntegrationTest.cpp and test classes, build and run for integration testing for your SPA
- **Empty SPA project:**
Fill in the code for your SPA
- **UnitTesting:**
Implement UnitTest.cpp and test classes, build and run for unit testing for your SPA

Running the AutoTester

- Running from the command line (cmd in Windows)
 - > `./autotester ../../../../tests/Sample_source.txt`
`../../../../tests/Sample_queries.txt`
`../../../../tests/out.xml`
 - Open output.xml in Mozilla Firefox to see the results
 - Note: analysis.xsl is in the same directory with output.xml
- Build and Compile using your platform instructions