

CS2212

Introduction to Software Engineering

Announcements & Project Info.



Design Documentation Due

- Was due on March 6th at 11:55pm.
- Make sure to submit it on OWL if you have not already done so.

Upcoming Due Dates

- **Implementation** and **Testing** components due the **week of April 3rd to 6th** based on the date of your acceptance testing with your TA.
- Your Implementation and Testing components must be submitted within 24 hours of your acceptance testing.
- You can use late coupons on the submission of these components on OWL but not for the acceptance testing meeting it's self.
- Keep in mind Implementation and Testing are two different components and each needs it's own late coupons, so to push both back by 2 days would take 4 coupons.

Upcoming Due Dates

- The **Project Management** component is due on **April 6th at 11:55pm**
- Need to submit PDF of all meeting notes and team contract.
- Only one team member needs to submit the PDFs.
- This component will also be marked based on your use of Bitbucket, Confluence, and Jira.
- See Project Management component on OWL assignments tab for more details.

Upcoming Due Dates

- The **Peer Review** component is due on **April 6th at 11:55pm**
- This is an individual component and each team member must submit their own copy.
- Must fill out an Excel sheet template provided in Peer Review component description (found on the OWL assignments tab).

Implementation & Testing




- Have about a month to implement the project and test it.
- What is expected is a prototype that fulfills the requirements in the project specification.
 - Does not have to be bug free or perfect but should have the bare minimum for each functional requirement and your one extra feature.
 - Non-functional requirements should also be considered such as having an easy to use UI (does not have to be beautiful, but should be useable).
- Should also contain:
 - JavaDoc comments for all public methods and classes. Do not need to document autogenerated code (e.g. from GUI builder).
 - Other traditional Java comments inside methods, etc. as needed.
 - JUnit tests for all public methods in core classes (not GUI).
 - Detailed README file that explains both how to build and use the software.

Implementation & Testing

Some Advice:

- In the past students have stated that it was very helpful to have everyone in one room working on the code. This allows the team to provide help to each other live and discuss any design changes immediately if a problem is encountered.
- Leave enough time to integrate and test your code. Do not start putting together your team members code right before the due date. Also allow enough time to merge branches and deal with conflicts.
- Focus on important functional requirements, don't get caught up in nice-to-haves, unneeded extra features, or GUI.
- It is ok if you have to make some design changes when implementing your project. This is normal and expected. You don't have to follow your design document 100%, but it should still be clear how you got from that documentation to your current design. You may go back and update the documentation on Confluence, but this is not required and will not be regraded.

Example NetBeans Project and GUI

 Overview Site Info Announcements Syllabus CS1 Tool Course Content Team Project

Project Specification

Project Software

Project Components

Project Resources

Team Rosters

 Forums TA Consulting Assignments Gradebook

Team Project

Add Content +

Reorder ⚙

[Team Project](#) > **Project Resources**

This page contains links to libraries, APIs, and tools you might find useful in your project. You are not required to use them. If you have come across more resources that you think other students would find useful, please let us know (we will give you extra points for this!).

Graphical User Interface (GUI)

It is recommended to use [Java Swing](#) to design your GUI, but using [JavaFX](#) is also allowed. An example Swing GUI created with NetBeans's GUI builder is included below. You are free to use this code in your project.

Example Swing GUI and NetBeans Maven Project

- **NetBeans Maven Project** (for use with the NetBeans IDE).
 - Contains a `pom.xml` file that includes most dependencies you will need (JSON libs, JUnit, etc.)
 - Contains example JavaDoc comments and a JUnit Test (found under `src/test`).
 - `GUIExample.java` has a simple Swing GUI related to the project.
 - `JSONDemo.java` has a simple demo of using the `org.json.simple` to read a JSON file.
 - `pom.xml` is set up to build a jar that includes dependencies.
 - If you will be using JavaFX you will need to configure the project for this, [see this link for details](#).

Example NetBeans Project and GUI

Projects xFilesServices

GIS_Project

Source Packages

com.cs2212.gis_project

GIS_Project.java

GUIExample.java

JSONDemo.java

Test Packages

com.cs2212.gis_project

GIS_ProjectTest.java

Other Sources

src/resources

data

example.json

imgs

poi.png

maps

mc0.png

mc1.png

mc2.png

mc3.png

mc4.png

Dependencies

Test Dependencies

Java Dependencies

Project Files

pom.xml

Hello World Example

(used in JUnit Example and has JavaDoc Comments)

Simple Swing GUI Example

(also shows loading images)

Simple JSON Example

(using org.json.simple)

Example JUnit Test

(tests GIS_Project.java)

Project Resources

(JSON files, images, etc.)

Apache Maven

pom.xml file

For details on Apache Maven see:
What is Maven:
<https://maven.apache.org/what-is-maven.html>

pom.xml:
<https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>

Swing GUI

Projects x

Files

Services

GIS_Project

Source Packages

com.cs2212.gis_project

GIS_Project.java

GUIExample.java

JSONDemo.java

Test Packages

com.cs2212.gis_project

GIS_ProjectTest.java

Other Sources

src/resources

data

example.json

imgs

poi.png

maps

mc0.png

mc1.png

mc2.png

mc3.png

mc4.png

Dependencies

Test Dependencies

Java Dependencies

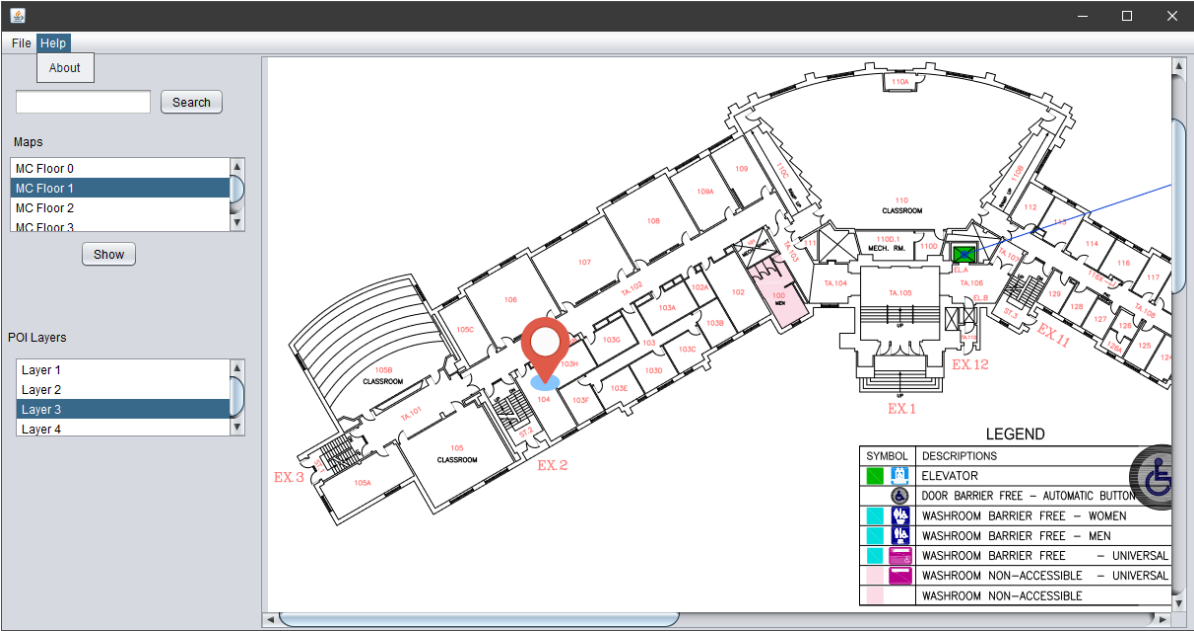
Project Files

pom.xml

1. Right Click on GIS_Project and select "Clean and Build"

2. Right Click on GUIExample.java and select "Run File"

Swing GUI Example



Example NetBeans Project and GUI

Notes:

- If you are using JavaFX some extra configuration is required, see: <https://openjfx.io/openjfx-docs/#introduction>
- pom.xml contains org.json, org.json.simple, and JUnit libs. Should download these for you once you run “Clean and Build”. If you are using other libs you will have to add these.
- You are free to use anything in this example in your project but you are not required to.
- Same pom.xml and files should work in other IDEs.